AD-A118 419 MASSACHUSETTS INST OF TECH CAMBRIDGE MAN-MACHINE SYS--ETC F/G 5/8
THE USE OF MACHINE AIDS IN DYNAMIC MULTI-TASK ENVIRONMENTS: A C--ETC(U)
JUN 82 W T WOOD N00014-77-C-0256

UNCLASSIFIED NL

1 OF 2

AD A118419

THE USE OF MACHINE AIDS IN DYNAMIC MULTI-TASK ENVIRONMENTS: A COMPARISON OF AN OPTIMAL MODEL TO HUMAN BEHAVIOR

WILLIAM T. WOOD

JUNE 1982

CONTRACT N00014-77-C-[illegible]
WORK UNIT NR 196-152
ENGINEERING PSYCHOLOGY PROGRAMS
OFFICE OF NAVAL RESEARCH
ARLINGTON, VIRGINIA 22217

82 08 20 [illegible]

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

REPORT DOCUMENTATION PAGE

READ INSTRUCTIONS BEFORE COMPLETING FORM

1. REPORT NUMBER	2. GOVT ACCESSION NO. AD-A118 419	3. RECIPIENT'S CATALOG NUMBER
4. TITLE (and Subtitle) The Use of Machine Aids in Dynamic Multi-Task Environments: A Comparison of an Optimal Model to Human Behavior		5. TYPE OF REPORT & PERIOD COVERED Technical Report Feb. 1, 1982-Jan. 31, 1983
		6. PERFORMING ORG. REPORT NUMBER
7. AUTHOR(s) William T. Wood		8. CONTRACT OR GRANT NUMBER(s) N00014-77-C-0256
9. PERFORMING ORGANIZATION NAME AND ADDRESS Massachusetts Institute of Technology Cambridge, MA 02139		10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS NR 196-152
11. CONTROLLING OFFICE NAME AND ADDRESS Engineering Psychology Group Office of Naval Research (Code 442EP) Arlington, VA 22217		12. REPORT DATE June 1982
		13. NUMBER OF PAGES 105
14. MONITORING AGENCY NAME & ADDRESS (if different from Controlling Office)		15. SECURITY CLASS. (of this report) UNCLASSIFIED
		15a. DECLASSIFICATION/DOWNGRADING SCHEDULE

16. DISTRIBUTION STATEMENT (of this Report)

Approved for public release; distribution unlimited

Accession For	
NTIS GRA&I	☑
DTIC TAB	☐
Unannounced	☐
Justification	
By	
Distribution/	
Availability Codes	
Dist	Avail and/or Special
A	

17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report)

18. SUPPLEMENTARY NOTES

None

DTIC COPY INSPECTED 2

19. KEY WORDS (Continue on reverse side if necessary and identify by block number)

man-machine systems	decision theory
man-computer interaction	workload
human behavior	manual control

20. ABSTRACT (Continue on reverse side if necessary and identify by block number)

Due to developments in the fields of computer and microprocessor technology, more and more tasks that previously only could be done by humans can now be assigned to machines. This study explores cases where such smart machines act as aids to human operators in dynamic multi-task systems. The operator must act as a supervisor and assign these machine aids to tasks he discovers. The operator also can do tasks himself.

(continued on next page)

An experimental paradigm was developed where the assignment behavior of subjects could be examined. Tasks that varied according to arrival rates, service times, rewards and costs appeared for service before an operator. The number of machine aids and the number of task classes were also experimental variables.

An optimal model was developed to provide a baseline against which to measure subject performance. The model forms a decision tree which looks N steps into the future. The branch that maximizes reward per time over the planning horizon was chosen as the best alternative. Subject performance was measured in terms of actual score and by the fraction of actual operator decisions that agreed with the model.

Subject performance was found to vary with the type of decision considered. When searching for tasks to deal with, subjects employed simple and sub-optimal heuristics. When faced with a decision to assign machine aids, subjects performed well when the machine had a clear productivity advantage. Unproductive machines, however, were used far more frequently than indicated by the optimal model. Increasing the cost of using machines was found to have a greater inhibiting effect on their use than did decreasing machine productivity.

THE USE OF MACHINE AIDS IN
DYNAMIC MULTI-TASK ENVIRONMENTS:
A COMPARISON OF AN OPTIMAL MODEL
TO HUMAN BEHAVIOR

by

WILLIAM TRUEMAN WOOD

B.S., Massachusetts Institute of Technology
(1979)

SUBMITTED TO THE DEPARTMENT OF
MECHANICAL ENGINEERING IN PARTIAL
FULFILLMENT OF THE REQUIREMENTS
FOR THE DEGREE OF

MASTER OF SCIENCE

at the

MASSACHUSETTS INSTITUTE OF TECHNOLOGY

June 1982

Signature of Author____________________________________
Department of Mechanical Engineering
May 24, 1982

Certified by Thomas B. Sheridan
Thomas B. Sheridan
Thesis Supervisor

Accepted by____________________________________
Warren M. Rohsenow
Chairman, Department Graduate Committee

ACKNOWLEDGEMENTS

I would like to thank Professor Thomas B. Sheridan for his supervision, advice and patience over the course of this project. I would also like to thank Professor Joseph Tzelgov of Ben Gurion University of the Negev, Israel, who provided an interested ear and many useful suggestions.

I owe much to my subjects who sat through so many hours of often tedious experiments, and without whom none of these results would have been possible. And of course I wish to thank the entire staff of the Man-Machine Systems Laboratory at M.I.T. whose own research provided countless distractions to my own.

I wish to acknowledge the support of the Engineering Research Programs, Office on Naval Research, Arlington, Virginia [Contract N00014-77-C-0256]. Gerald Malecki was the contract monitor.

TABLE OF CONTENTS

Page

1. INTRODUCTION....................................1

1.1 The Rising Capabilities of Machines.....1

1.2 The Man-Machine Interface...............1

1.3 Motivation for Studying the Conceptual Interface..............................2

1.4 The Role of the System Designer.........4

2. MODELING MAN/MACHINE SYSTEMS..................5

2.1 Defining Man-Machine Systems............5
2.1.1 Examples of Man-Machine Systems of Interest
2.1.2 Examples of Man-Machine Systems Not Dealt with in this Study

2.2 Characteristics of Man-Machine Systems..7

2.3 Characteristics of the Machine Aids.....10

2.4 Characteristics of the Task Environment.10

2.5 Normative Models of Human Tasks.........12

3. QUEUING THEORY AND THE MODELING OF MULTI-TASK ENVIRONMENTS........................13

3.1 The Language of Queuing Theory..........13

4. OTHER WORK EXPLORING MULTI-TASK ENVIRONMENTS..15

4.1 The Tulga Paradigm......................15

4.2 Models Based on Queuing Theory..........16

5. CYCLIC GAME WITH DISTINCT WORK AREAS..........17

5.1 The Experimental Paradigm Used in This Study...................................17

5.2 Operator Duties in SUPER................21

Page

5.3 Building the Normative Model of SUPER...22
5.3.1 The Decision to Change Work Areas
5.3.2 The States in a Two-Identical-Work-Area / No-Machine Scenario
5.3.3 State Probabilities
5.3.4 Expected Reward/Time of an Operator's Strategy
5.3.5 Determining an Optimal Strategy

6. UTILITY IN AN ENVIRONMENT OF CERTAINTY........33

6.1 Linear Utility Assumption...............33

6.2 Testing the Linear Utility Assumption...34

6.3 Measuring Operator Performance..........37

6.4 Experimentation.........................40

7. THE OPTIMAL DECISION MODEL....................43

7.1 Limitations of the Markov Model.........43

7.2 The Simulation, or Monte Carlo, Approach................................47

7.3 Using a Decision Tree to Generate Optimal Stategies.......................48

7.4 Extensions to the N-Step Model..........54

7.5 Pruning the Decision Tree...............55

8. MODEL VALIDATION............................59

8.1 Comparison to the Markov Model..........59

8.2 Comparing Various Levels of the N-step Search...............................60
8.2.1 Comparision by Reward per Time
8.2.2 Comparison by Decision Type

9. ANALYSIS OF HUMAN BEHAVIOR...................63

9.1 Measures of Human Performance...........63
9.2 Some Reasons for Operator Sub-optimality.........................65
9.2.1 Non-Linear Utility

Page

9.2.2 The Operator's Internal Representation of Task Parameters
9.2.3 The Mental Computation Stage
9.2.4 Operator Reaction Time

9.3 The Experimental Situation..............70
9.3.1 The Task Familiarization Phase
9.3.2 The Warm-Up Phase
9.3.3 The Experimental Phase
9.3.4 The Debriefing Phase

10. EXPERIMENTAL DESIGN..........................75

10.1 Choosing Task Parameters 75
10.1.1 Determining Satisfactory Characteristic Time Interval

10.2 Finding Instances of Human Sub-optimality.........................79

11. EXPERIMENTAL RESULTS.........................81

11.1 Introduction...........................81

11.2 The Operator's Search Strategies.......81
11.2.1 The Effect of Transition Time
11.2.2 Human Information Processing Limitations
11.2.3 The Effect of Subjective Probability Estimation

11.3 Choosing How to do Tasks...............84
11.3.1 Machines That are More Productive Than Their Supervisors
11.3.2 Men and Machines with Comparable Abilities
11.3.3 Assigning Low Productivity Machines
11.3.4 The Impact of Machine Wages on Strategy

11.4 Conclusions............................93

12. CONCLUSIONS AND SUGGESTED FURTHER RESEARCH...95

12.1 Conclusions From This Study...........95

12.2 Possible Extensions on this Study......96

13. REFERENCES.................................97

Page

APPENDICES

A. MATHEMATICS OF POISSON PROCESSES.............101

A.1 Poisson Processes.....................101

A.2 The Exponential Distribution..........101

A.3 Formulae with Holding Cost............102

B. LISTINGS OF COMPUTER PROGRAMS...............103

LIST OF FIGURES

Figure		Page
2.1	Man/machine system characteristics	9
5.1(a)	Photograph of the SUPER playing field	18
5.1(b)	Diagram of the SUPER playing field	19
5.2	The control box	20
5.3	Task parameters for an environment with two work areas and no machine aids	24
5.4	State transition diagram for the task environment with two identical work areas and no machine aids	28
5.5	The optimal value of PC as a function of the parameters in the Markov model	31
6.1(a)	Photograph of the BOXCLR playing field	35
6.1(b)	Diagram of the BOXCLR playing field showing parameters x(i) and "a" used in analysis	36
6.2	Sample histogram showing the distribution of strategy costs in BOXCLR for the case of eight boxes and a machine wage of four points per second	40
7.1	The reward per time observed at various points in a simulation of SUPER	46
7.2	Options for an operator at each node in a decision tree, for a paradigm environment with M machine aids and R work areas	50
7.3	The 3-step decision tree for an operator faced with two work areas and having one machine aid	53
9.1	Representation of a subject encoding task information for use in his decision algorithm ultimately leading to action	67
9.2	Movement from task parameters to operator action	69

Figure		Page
9.3	Megatek display during the Task Familiarization Phase of experimentation	71
10.1	Photograph of subject absorbed in an experiment with SUPER	76
11.1	Operator performance in assigning low productivity machine	87
11.2	Fraction of all decisions to deal with a task where the low productivity machine was used	89
11.3	Operator performance in assigning an expensive machine aid	91
11.4	Fraction of all decisions to deal with a task where the expensive machine was used	92

LIST OF TABLES

Table		Page
6.1	Fraction of all strategies better than the one used by subjects	41
6.2	The ratio of operator cost to minimum cost for experimental subjects	41
7.1	Required states for representing the R work area, L time division, no machine environment	45
7.2	Time required to estimate RPT to various levels of accuracy	48
7.3	The number of branches in the decision tree looking at R work areas N steps into the future	56
8.1	Average ratio of N-step decision model RPT to Markov optimal RPT	59
8.2	Typical reward per unit time as a function of the depth in an N-step search	60
8.3	The number of times a M-step model agreed with a N-step model when all decisions are considered	61
8.4	The number of times a M-step model agreed with a N-step model when only decisions to assign machine aids are considered	62
10.1	Average performance relative to an optimal model for three subjects	78

CHAPTER 1

INTRODUCTION

1.1 The Rising Capabilities of Machines

Computers are being developed that can handle more complex tasks than ever before. A major justification for the development of intelligent machines is that it frees human beings to do other tasks that are potentially more enjoyable and fulfilling. Economic justification lies in the increased productivity of the machine-aided human.

As the number of intelligent machines increases, however, a workforce capable of directing and supervising these machines will be needed. The shortage of computer programmers in modern society shows how the growth in applications for a new machine (the computer) can outpace the growth of workers capable of administering it. As machines become less specialized and can be applied to a variety of problems, more and more jobs will involve direct interaction with machines. Either the method of interaction must be simplified or the people in these positions must be trained.

With machines performing a larger fraction of tasks in society, fewer human beings will have occupations dealing directly with the environment. Instead people will have jobs supervising machines. Problems are bound to arise in the interface between the man and the machine.

1.2 The Man-Machine Interface

The interface between man and machine can be examined on two levels. The first level of interaction is the physical interface, involving the specific command vocabulary, the number of buttons that must be pushed, how they are arranged, and how the machine communicates with the

operator. The rate of information that can be transmitted between a human and a machine aid will determine just how effective the man/machine system will be. The more transparent the physical interface, the more the system can accomplish.

There is another level to the man/machine interface that is perhaps more important if only because it has been less extensively examined than the physical interface. This second level is the cognitive interface. When a man/machine system is designed, a decision is made to apportion the components of a task into two segments: a segment that will be handled by machine and one that will be handled by man.

The points where the duties and responsibilities of the human interact with those of the machine can be termed the cognitive interface. While the physical interface can be defined solely in terms of the hard- and soft-ware components of the man/machine system, the conceptual interface must include a consideration of the nature of the task to be performed, and the method used to deal with it.

The cognitive interface is like the storm front between a warm air mass and a cold air mass. Both are described as areas of interaction rather than specific lines. When someone is near the boundary, it is hard to say just where it is. Like a storm front the cognitive interface is a region where confrontation (i.e. bad weather) is most likely to occur; hence it is interesting to study. However, because it is a region and not a point, and because it is dependent of the nature of the tasks it is hard to come to any general conclusions. But it can be done.

1.3 Motivation for Studying the Conceptual Interface

Computers are very good number crunchers. They unerringly follow complex instructions and they think very fast. Humans, on the other hand, tend to be innovative and can deal with unusual problems taking into account such criteria as "the political reality of the situation" and the risk to human life that cannot be well defined. The designer of man machine systems should try to exploit the relative strengths of men and machines in formulating a system.

Unfortunately, there is no guarantee that a system that is based soley on the strengths of its components will itself be strong. It is equally important to consider how the components will interact and communicate. Physical structures seldom fail due to component failure. Problems usually arise in the joints between members. Failure due to component interaction and communication arise in many diverse systems.

Consider the United States and the U.S.S.R as possible partners in the creation of a new economic order. With agreement on arms control both nations would be able to realise greater economic growth, improved standards of living and still be able to give more economic aid to lesser developed countries. Institutions such as the United Nations and the Club of Rome are trying build a future based upon such cooperation between member nations of the world. However, such cooperation is highly unlikely due to the fear and the lack of trust which exists across national boundries. Systems which suppose that the United States and the Soviet Union will act together, even for mutual benifit, neglect the psychological barriers that prevent the components from effective communication.

Many systems designed to exploit the strengths of components have failed because of the interaction between components. An example that pertains more to the idea of man-machine interaction involves the use of robots on certain assembly lines. An assembly procedure involves performing a long sequence of tasks on a product. Only some of these tasks can be readily automated. When industrial robots are developed they replace human workers on the assembly line.

This strategy will often lead to human workers surrounded by machines. The human takes a component from a robot, performs a task, then gives the part to a second robot. The lone human is isolated from the world of flesh and bones. He is paced by the machine feeding parts to him and he loses his sense of identity; he does not enjoy his work and his productivity falls. The moral of this story is that placing robots in jobs that they were best equipped for does not always ensure an efficient system.

A similar example is the decision by many small businessmen to buy a small computer for their companies. They feel that the computer will be able to solve their accounting problems, reduce the load on their secretaries and allow for better planning. These are all tasks that small computers are able to do well and which will free other employees to do more productive work.

Unfortunately, the computer often meets substantial resistance from the workers who will have to use it; often they are "scared" of it. They don't know how to use the computer, are unwilling to learn how, and worried that it will replace them. The employees find excuses not to use the machine which ultimately ends up discarded in the stock room.

1.4 The Role of the System Designer

When a system designer is constructing a man/machine system, he has some conception of where the conceptual interface will lie. In deciding which aspects of a task will be handled by the human and which can be handled by the machine, the designer makes various assumptions concerning the capabilities of the components. Given a choice of system configurations, the capable designer will try to find an optimal system. In doing this he will make various assumptions concerning how the components will interact. If he makes unrealistic assunptions he is likely to make sub-optimal decisions.

In the previous example, the businessman who sought to improve productivity played the role of the system designer. While he hoped to exploit the strengths of automation he neglected to account for the hostility the machine encountered from his employees. Consequently, a poor decision was made.

The physical interface has been extensively explored in much of the human factors literature. Methods of conveying information, either commands or reports, between the man and the machine have been developed and the system designer can draw on these developments in his design.

The cognitive interface, however, is still far less documented. Deciding how the man and machine are supposed to interact in dealing with the task at hand is left to the designer's intuition. Deciding how the components will in fact interact is left to the human operator as he becomes familiar with his job. If supposition and fact do not coincide, the resulting system will probably be very inefficient.

This study will look at the cognitive interface and give some indication to the system designer as to how humans can be expected to interact with machines. Suppose there are consistent biases in the way operators deal with machines. The designer will want to avoid placing the operator in situations where these biases will prevail lest the system behave too sub-optimally.

The literature if full of examples of the different reasons humans act sub-optimally. Questions of utility, of risk aversion and of perception have all been discussed. This report, however, is not a study of how human goals and perceptions of rewards and costs may differ from those assumed by a system designer. Rather it is an exploration of the pyschological barriers and biases that exist in human operators that hinder effective man-machine communication.

CHAPTER 2

MODELING MAN/MACHINE SYSTEMS

In order to explore the impact of the man/machine interface on the performance of man/machine systems, some benchmark is required with which to compare this performance. A natural baseline is "optimal" performance. The term "optimal" does not refer to the best way of dealing with the problem that the man/machine system is designed to handle. Rather it refers to the best strategy that the human operator can follow acting under the established constraints of the man/machine configuration. A strategy is defined as best if it maximizes a given objective function or criterion for performing the task.

2.1 Defining Man-Machine Systems

The term "man/machine system" can be used to describe anything from a child playing with a electric train to a space shuttle mission. In the strictest definitional sense, all that is required is a human and a machine acting towards a common goal. The man and the machine do not even have to be aware of each other's existrnce, as is the case in the telephone company where human and computer operators work side-by-side each handling a specific subset of calls. Additionally, there need be no mention of responsibility, authority, or the chain of command, in the pure definition.

The analysis in this report does not pretend to consider all variations of man/machine systems, instead focusing on a particular subset of interest. This subset is broad enough to include an area of general interest to the system designer.

The concern in this report encompasses those applications of man/machine systems where a human is faced

with a variety of tasks that must be accomplished. These tasks might include writing a letter or maintaining the flight path of an airplane. Potentially the human could complete all the tasks manually. However, the tasks that must be accomplished may arise fast enough that the operator's ability is strained. In some cases he does not have enough time to do all that is required. Consequently, the human might be supplied with a machine aid capable of undertaking many of the tasks faced by the system thereby reducing the stress on the human operator.

While the machine (or computer) may not be able to do everything the human does, it can handle many tasks satisfactorily. Some tasks the machine will handle better or faster than the human.

Computers were first used in application that exploited their quick number crunching abilities including the dull tedious jobs that workers previously were burdened with. These first applications resulted in computers replacing men. This project deals instead with cases where humans remain in the system but their productivity is enhanced through machine aids.

The concern here is with instances where the human is given the responsibility for maintaining system performance and the authority to direct the components of the man/machine system. The human acts as a supervisor directing the allocation of both his time and the time of any machine aids subordinate to him. The focus of this study is on this allocation.

2.1.1 Examples of Man-Machine Systems of Interest

Modern aircraft are outfitted with flight computers that could feasibly take a plane from Detroit to San Diego without there even being a pilot in the plane. A pilot could perform the entire flight himself if he so desired. This latter option might cause a bit of strain on the pilot because he has other requirements on his time, such as talking to ground control, informing the passengers of turbulence and prominent landmarks, and watching out for other planes in his airspace. It is no surprise that the pilot makes use of his flight computer and collision warning systems. However, it is always the pilot's option to use these aids. Ultimately he is responsible for the lives of everyone on board.

The small businessman, with his new personal computer, is another example of the type of man/machine system considered in this paper. Various requirements on a businessman's time crop up over the course of a day. Letters must be written, sales forecasts must be made,

inventory must be maintained, and employees must be paid. The personal computer was bought to help the company deal with all these tasks, yet it can only do one thing at a time. The manager must decide which jobs have priority on the computer and which he must do himself.

With the continued expansion in micro-processor applications, many more jobs will become automated. The potential for machine aids in a variety of jobs is immense. Someday entire factories will be automated. Only a skeleton supervisory work force will be necessary to plan production runs. These people will allocate the resources of the plant based on current orders and anticipated demand.

2.1.2 Examples of Man-Machine Systems Not Dealt with in this Study

There are many applications of man-machine systems that do not fall into the category considered in this report. Purely automatic systems are one example. These systems are turned on and then work continuously. There is no operator. If failures occur then the machine itself must cope.

A household heating system is an example of an automatic system. Once it is activated it relieves people of the responsibility for regulating temperature. The system works continuously and does not have to be reassigned to the task of adjusting temperature every time temperature needs to be adjusted. Once activated there is no interaction between the machine and the humans it serves.

At the other end of the spectrum are machines that require constant human supervision. A lawn mower, and other household tools, make tasks easier for the human operator. However, they cannot simply be assigned to a task and then left while the operator deals with something else. These tools instead require constant interaction with the operator in order to function properly.

2.2 Characteristics of Man/Machine Systems

In modeling man/machine systems it is important to reflect the important characteristics of these systems. In these systems a human operator is faced with a variety of tasks that require attention. The operator is also given one or more machine aids whose work he supervises. It is realistic to assume that the operator seeks to maximize some reward function as he works.

For simplicity, a unidimensional reward function will be considered. In life, it can be argued, humans try to maximize their acquisitions of a variety of attributes

including money, happiness and vacation time. But these are global concerns in their life as a whole.

For individual pursuits, such as games or work, it is an acceptable approximation to assume that their tactical decisions are based upon maximizing a single objective. While a human may participate in sports for his health and well being, his actual play is governed by his desire to maximize his score. He may choose his occupation for many diverse reasons, but his goal on a given project is to maximize profits. For the purposes of analysis this report will assume a human operator in a system seeks to obtain as much of some attribute as possible, whether this attribute be money, or "utility".

In the type of man/machine system considered, the man decides whether to do the tasks himself or whether to assign a machine aid to them. We can divide the tasks faced into N classes. Classes are distinguished by the frequency (probability) of occurance; how fast they can be dealt with; and by their effects on the reward function.

We can assign each task class i a mean arrival rate L(i) [arrivals/unit time]. This corresponds to a mean arrival time of 1/L(i).

Task arrivals will be assumed to havea Poisson distribution. A Poisson process can be totally specified by the parameter L(i). Poisson processes are more fully described in Appendix A.

A second important characteristic of tasks is the time required to deal with them. A service rate u(i,j) [tasks served/unit time] represents the mean number of tasks of class i that can be completed per unit time. The subscript j is present to indicate whether a man or a machine is servicing a task. The case of multiple machine aids, having various abilities is a simple extension of this case.

The service time can be either deterministic or probabilistic. To simplify the function u(i,j) the effect of task class i and servicer class j can be separated. One way is to assume a heirarchy of abilities such that if a machine is faster at one task than a human, or another machine, it will be faster at all classes. Mathematically, this assumption would take the form:

$$u(i,j)=u(i)*c(j)$$

where u(i) is independent of the server, j, and the factors c(j) are independent of task class, i.

Costs and rewards must also be specified in a complete

characterization of a system. We can define R(i) as the reward obtained for completing a task. There may also be a holding cost h(i) that is incurred as long as a task has not been attended to.

The holding cost represents a penalty for putting off completion of a task. In reality there is usually an incentive to finish one's work as early as possible. Leaving jobs to the last minute means forgoing unforseen opportunities that arise. In financial matters all costs and rewards are discounted to their net present value. Rewards achieved in the future are not as valuable as the same rewards achieved today. The inclusion of a "holding cost" in modeling man/machine systems is a simple way of representing the penalty for procrastination.

Holding cost does not behave exactly like a discount rate. For the latter, the value of a particular task would decay exponentially down to zero the longer its undertaking was put off. For a holding cost, the value would drop linearly and would become negative if a task remained unattended for a sufficiently long period.

Another cost in man/machine systems is the cost of using machine aids. Using machines is not free. There are energy costs, depreciation, rental fees if the machine is not owned, and other operating expenses. These costs can be summarized in a "wage" for the machine help. Every time a machine in class j is used it incurs a cost w(j) per unit of time. Avoiding this cost is one reason why an operator might be hesitant to use machine aids.

h(i)	: the holding cost of leaving task i unattended (cost/time)
R(i)	: the reward for completing a member of task class i (reward units)
w(j)	: the cost, or wage, of operating a machine of type j (cost/time)
l(i)	: the mean arrival rate of tasks in class i (arrivals/time)
u(i,j)	: the mean service rate of tasks in class i by machines in class j (tasks/time)

Figure 2.1 - Man/machine system characteristics

The reward function that the human operator seeks to

maximize will be a function of the system variables summarized in Figure 2.1. The exact form of this functional relation will depend upon the structure of the man/machine interaction and the tasks the operator faces.

2.3 Characteristics of the Machine Aids

One structural question is whether the machines are general purpose or whether they are designed for specific tasks. In an aircraft cockpit, the pilot is supplied with individual flight computers for heading, for pitch control and for speed, as well as a collision avoidance system. Each of these mechanical aids is dedicated to a particular task and cannot be used for other than its primary function.

A computer, on the other hand, is able to do a variety of tasks. In companies, the computer can print cheques, type letters, and record sales: it is a general purpose machine aid. The difference between dedicated and general machine aids is reflected in the parameter u(i,j), but it is important to remember there is a fundamental difference between the two modes of system design.

An important consideration is the autonomy of the machine from the human supervisor. At one extreme are machines that do not interact with humans at all, such as the welding robots in factories which consistently perform a whole series of assigned task with little or no human intervention. At the other end of the spectrum are machines like pencil sharpeners and lawn mowers whose actions must be continually directed by a human operator. These extreme cases are not of interest in this project; the former because their is no man/machine interaction to study, and the latter because they are machines that merely increase the productivity of a human laborer without changing his basic strategy for attacking the problem at hand.

Between the extremes, however, there are many types of man/machine systems. Machines that are capable of performing a short task, or sub-task, but cannot perform a whole series of such tasks without being reprogrammed for each are of interest in this study. The assignment process can be complex, as in the case of computers that must be programmed before they can attack a problem, or very simple, as in the case of a pilot switching on his autopilot.

2.4 Characteristics of the Task Environment

Because the design of a man/machine system is heavily tied to the type of work the system will undertake, a complete description of the system must include characteristics of the "task environment". There are basic differences between the type of task performed by an

automobile driver or aircraft pilot, and by secretaries or computer programmers.

In the case of pilots, and other vehicle drivers, the operator normally responds to problems that arise. Airspeed, direction, cabin pressure and fuel all must be maintained within some acceptable levels. Actions must be taken only when indicators fall outside normal limits. When not actively correcting some problem, the pilot scans his instruments.

In this type of task environment the number of classes of tasks, or emergencies, which arise is finite. Also, when one problem arises, an identical problem cannot arise until the first is fixed. This type of task environment is called "cyclic". The operator cycles from one potential problem area to another dealing with tasks as they arise.

The complementary type of task environment, one which is faced by computer progammers or secretaries, have infinite queues. More than one task from the same class can appear to the operator, and a backlog of work, called a queue, can arise. In addition to infinite queues there are cases where arrivals "balk", or leave because of a long expected wait. The cyclic environment can be considered an extreme case of balking where no new tasks arrive if there is even one action-evoking event waiting for processing.

Another important characteristic of the task environment is the degree to which the machine aid operates in the same manner as the human. If both attack similar problems in a similar manner then they can probably trade off working on the same task. The man and the machine are interchangable. If one starts a task, the other can finish it.

Flying an airplane is an example of an interchangable man/machine system. The pilot can turn the flight computer on and off at will. He can relieve the machine with no loss of system performance.

At the opposite extreme is computer programs. Once the computer has started working on a problem, it cannot be replaced by a human. The computer and the man work at different speeds and with different approaches. When one has started working on a task it can only be replaced if the operator is willing to discard work already done and start over.

In task environments where man and machines are interchangable, the service times for tasks are approximately the same. When they are not interchangable, it is often because computers perform certain functions much

faster than humans can.

2.5 Normative Models of Human Tasks

When modelling man/machine systems to find some measure of optimality against which to compare observed performance, it is necessary to make certain simplifying assumptions about human beings as decision makers. A human working in complex environments may get flustered. Further, he may forget what he has seen and done previously. His decisions take a finite time to make.

An optimal model of an essentially human task cannot incorporate these human frailties. First, these frailties are unpredictable in nature, and second, they make the model sub-optimal. If they were included there would be nothing to prevent a person less prone to these weaknesses from coming along and outperforming the supposedly "optimal" baseline model. For the sake of analysis, the operator in the normative model will be assumed to react infinitely fast to changing circumstances, and to have a perfect memory of his past actions.

Another concern in the computation of an optimal policy is the time horizon that the human operator is concerned with. With an infinite time horizon the human will try to maximize his steady state performance. A word-processer operator who faces the same sorts of problems every day is in a steady-state situation. Other task environments have finite time horizons. A pilot knows that a given flight will end after a specified time and his behavior may change depending on where in his flight plan he finds himself.

CHAPTER 3

QUEUING THEORY AND THE MODELING OF MULTI-TASK ENVIRONMENTS

3.1 The Language of Queuing Theory

Queuing theory provides a useful vocabulary for representing multi-task situations. In queuing theory, customers appear before a service center, where one or more servers are located. If all servers are occupied the customer enters a queue and waits for service. When a customer is served, he leaves the queue and frees a server to deal with other customers.

Man/machine systems in multi-task environments can be considered in an analogous manner. Tasks requiring processing appear before the man-machine system, where either the man or his machine aids will deal with them. If neither is available to "serve" the task it enters a "queue" and waits for processing.

The above description presents queuing theory in its simplest form and provides a generic definition of what a queue is. However, many real-life queues have special features which differentiate them from the basic model.

One common feature in queues is customer balking. If a person arrives at a bank and finds he will have to wait for service he may decide to just walk out. This behavior is called balking. Balking can alse describe the enforced, as well as the voluntary, turning aside of customers. Queues often have a capacity limit. Once this capacity is reached, new arrivals are turned away unserved. A waiting room may be able to accomodate only N people. If all chairs are filled, additional customers are not welcome.

Many man/machine systems can be represented as queues with balking. If a task arrives in front of an operator who

feels overloaded, then the operator may simply decide to forget about the task and let someone else handle it.

In an airline cockpit, a pilot responds to discrepencies between his instrument readings and their desired levels. The appearance of discrepancies corresponds to customer arrivals. The queue for discrepencies in aircraft heading can only contain one "customer" for the following reason: the heading can be either correct or incorrect. If the heading is incorrect additional turbulence may increase its deviation from its desired state. However the operator only has to deal with heading once regardless of the magnitude of the discrepency. As far as the pilot is concerned there is only one task in the queue requiring service.

Similar to balking is the customer option to "renege". If a customer is forced to wait for too long he may get fed up and leave the queue, or renege on his implied request for service. In a multi-task environment, certain tasks may crop up that must be dealt with within some time window. A secretary may have to finish typing a report before a deadline. If the deadline is missed, the report no longer needs to be typed and the task leaves the queue.

In man/machine systems it is useful to consider queues with multiple servers. Tasks enter a queue. They can be served either by the man or by his machine aids. In systems considered in this report, where a human operator assigns tasks to the machine aids, the service facility is said to have multiple levels. First tasks are pre-processed by the human supervisor. Tasks then are processed by the machine aids.

When there is a situation of many task classes it is convenient to consider each task class as having its own queue. If one task class is to be given priority for service over other task classes then those other tasks will not be served until the high priority queue is completely empty.

CHAPTER 4

OTHER WORK EXPLORING MULTI-TASK ENVIRONMENTS

4.1 The Tulga Paradigm

To a major degree, this study is an outgrowth of the work done by Kamil Tulga exploring human decision making in a dynamic multi-task environment (see Tulga [24,25]). In Tulga's paradigm, subjects were faced with the problem of dealing with tasks appearing on a variety of work areas. As tasks were completed the subject's score would increase, and tasks left unattended would eventually disappear. The time before this disappearance, the time required to do a task, and the score for tasks varied from task to task.

Human behavior was compared to an optimal algorithm. This algorithm determined the order for dealing with all current tasks which maximized expected reward. The optimal ordering, or best path, was executed until a new task appeared at which point the strategy was reevaluated.

Tulga also showed how with specific simplifying assumptions, his algorithm could be reduced to encompass Job-Scheduling and Travelling Salesman problem. The algorithm uses dynamic programming and some ideas from Queuing Theory.

Results of Tulga's experimentation include that humans do not discount future returns as much as they should. The paradigm also provided a measure of subjective Work Load, or stress, felt by the operator.

The paradigm considered in my study is similar to Tulga's in that human subjects are faced with a dynamic multi-task environment. My study differs from Tulga through the addition of machine aids to help the operator deal with tasks. Where Tulga's focused on the way humans ordered

tasks for processing, my study focuses on the manner they choose to do those tasks.

Another important difference between the two studies is that Tulga's subjects had perfect information about the system. That is, they could see all available tasks and could instantaneously tell when new tasks arrived. In this study only one work area is visible at a time. The operator, while working on one task, cannot tell what else is expected of him, though he can infer rough probabilities that other task exists elsewhere based on their past observed arrival rates.

4.2 Models Based on Queuing Theory

In the Queuing Theory and Operations Research literature there has been some work in multiple server queues. Because humans and machine aids can be modelled as multiple servers, the problem of scheduling customer arrivals into a mulitple-service structure is the same problem faced by a human operator allocating tasks to his machine aids. Unfortunately, most of the operations research work in multiple-server queues is theoretical. Very little work has been done experimentally with human-machine interaction.

One exception to this statement is the work of William B. Rouse [17]. He has looked at situations where humans and computers have overlapping capabilities and responsibilities. Both the human and the computer are assumed to scan some display looking for tasks that require attention. When such an action-evoking event is found it is dealt with.

In simulation experiments the level of human computer feedback, the probability the human will make errors, the probability the computer will make errors and the relative productivities at doing tasks for the man and machine were all varied. Ways of allocating responsibility between the man and the machine to maximize performance are discussed.

Rouse's experiments differ from the ones in this study in two important ways. First, most of his experiments are simulations and do not use actual subjects. Second, his machine aids are highly autonomous. They search for action-evoking events and attempt to complete them without human intervention. In this report, the human operator must do all searching and then assign machines to tasks if he so decides. However, Rouse's task environment, where action-evoking events must be searched for, is very similar to that used in this study.

CHAPTER 5

CYCLIC GAME WITH DISTINCT WORK AREAS

5.1 The Experimental Paradigm Used in This Study

In the Tulga paradigm, subjects were able to examine all work areas with a single glance at any time over the course of an experiment. In most real life situations, a human being must gather information sequentially from various sources. In the extreme, tasks that must be done are located in physically different locations. When a businessman is working at his office, he won't be able to find out about or respond to a broken faucet at home.

Additionally, even when tasks arise in the same location, an operator may be so preoccupied doing one task that he will not notice new task arrivals. In the experimental situation, however, the actual act of doing a task is not intellectually taxing. An operator can scan other work areas and plan future strategy when he is supposedly "doing" a task.

To simulate the mental and physical separation of the operator from all but the task he is doing or the work area he is looking at the following experimental paradigm was created. A diagram of the playing field is presented in Figure 5.1. Figure 5.2 is a diagram of the control input box given to the operator.

In this game there are R classes of tasks that can possibly arise. For simplicity, only one member from each task class can appear at any time. Each task class arises in its specific work area. Only one work area is displayed at any time. If a task exists in a work area, it is signified by the display of a box on the work area. To complete the task the operator must move the task box to the right end of the work area. The operator can move the task

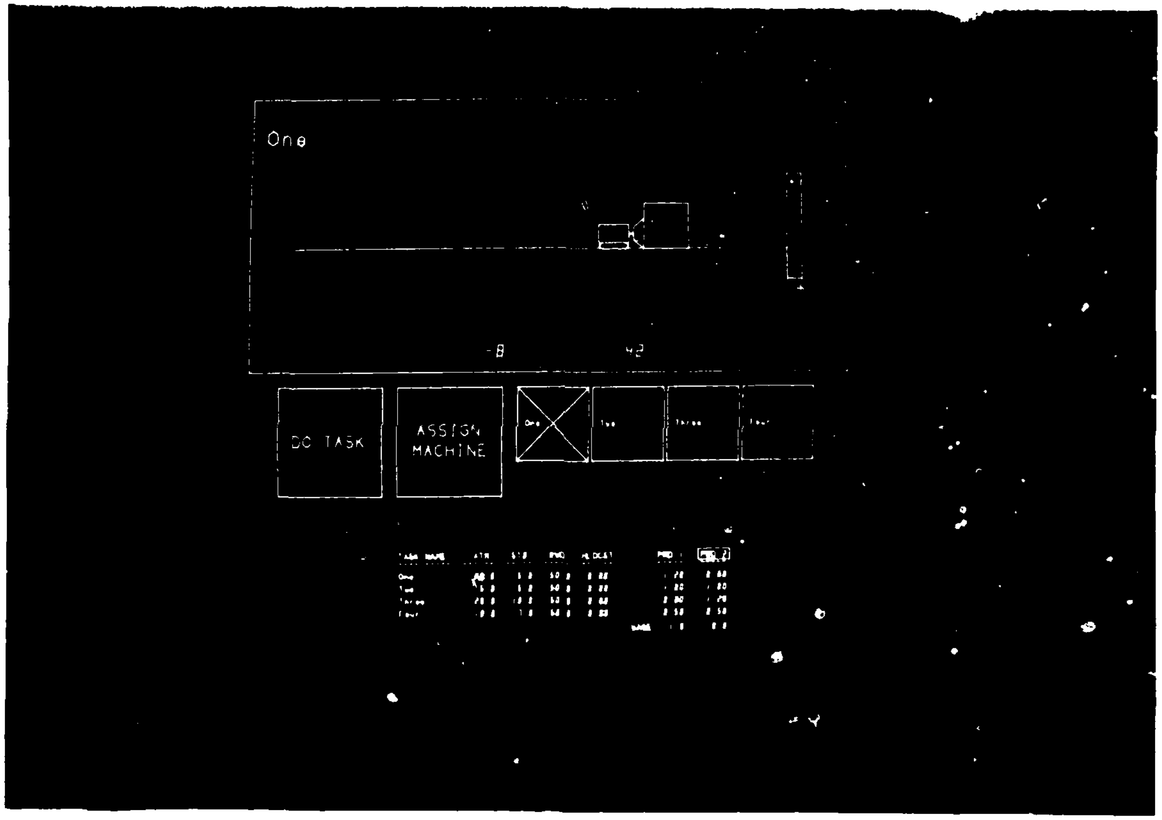

Figure 5.1(a): Photograph of the SUPER playing field

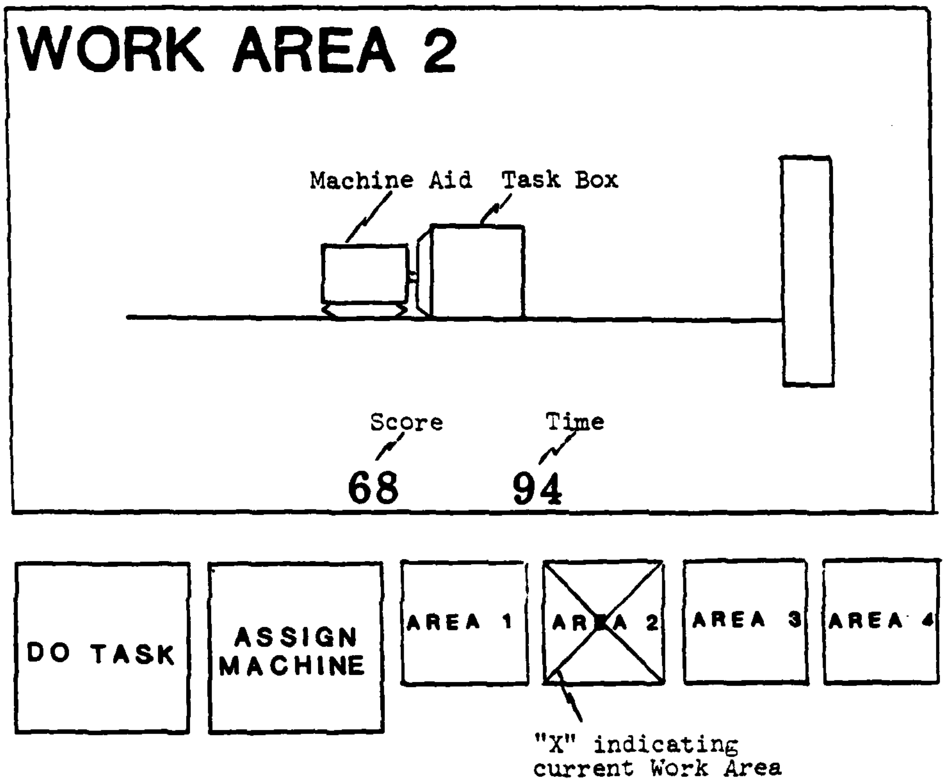

Figure 5.1(b): Diagram of the SUPER playing field

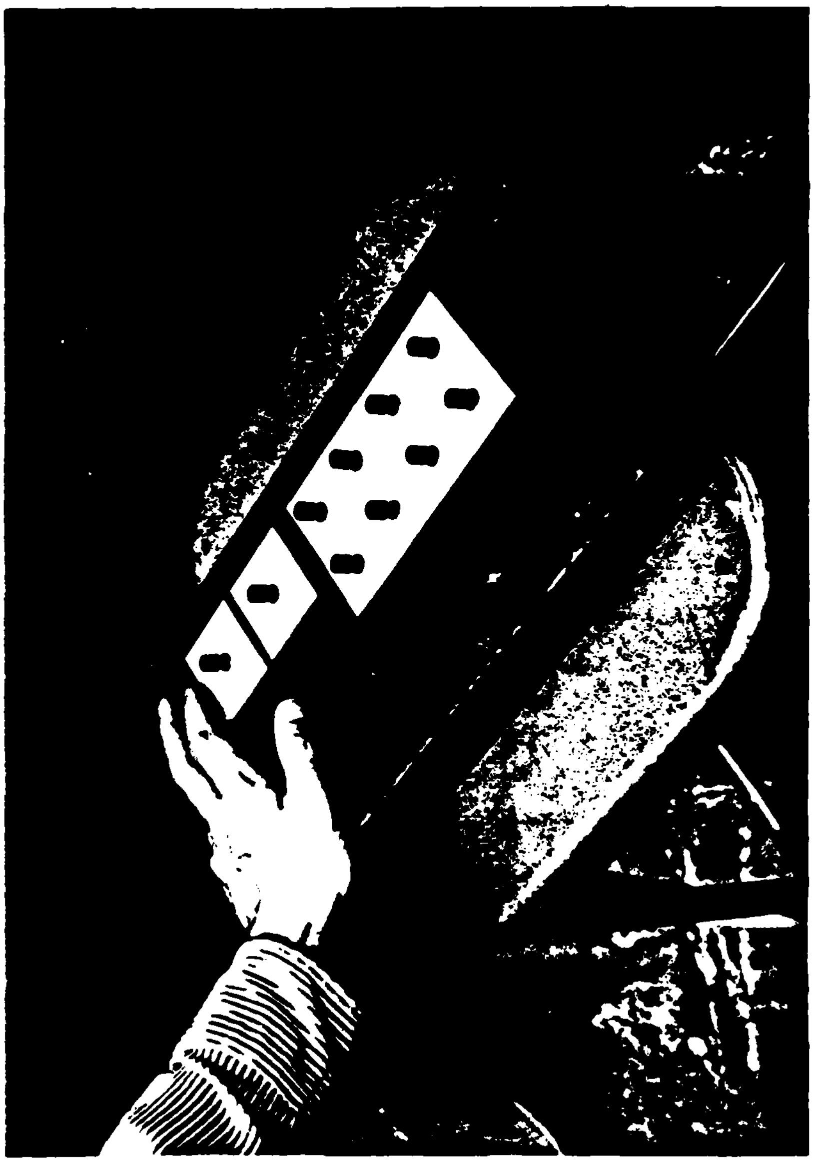

Figure 5.2: The control box

box currently displayed by pressing the "DO TASK" button on the control box. Alternately, he can assign a machine to do the task by pressing the "ASSIGN MACHINE" button.

The operator also has the option of changing the work area displayed. He makes this transition by pressing the button on his control box corresponding to the desired work area. A large X on the playing field indicates the current work area displayed (see Figure 5.1(b)).

The paradigm is entitled SUPER because the operator takes an essentially supervisory role over the machine aids.

The key elements in the role of an aircraft pilot are reflected in SUPER. The pilot scans his control panel looking for indicators which vary from their desired levels. If a discrepency is found, he can either correct it himself or assign his "machine aid". (This machine aid would be his flight computer, though the co-pilot could also be considered a machine aid.) For the pilot, changing work areas corresponds to his shifting attention amongst the various displays. Assigning the machine aid corresponds to the pilot giving instructions to his flight computer, or his co-pilot.

5.2 Operator Duties in SUPER

In SUPER the operator is faced with R work areas, and is given the aid of M machines. The task class in each work area has associated parameters for mean arrival rate, mean service rate, and reward and holding cost as described in Chapter 2.

The operator can examine only one area at a time. If a task exists on the displayed work area the operator has the option of doing the task himself (i.e. pressing the "DO TASK" button), assigning a machine to do it, if one is available (i.e. pressing the "ASSIGN MACHINE" button), or doing nothing. Of course the operator always has the option of swicthing to a new work area to attend possible tasks there.

An operator might want to switch to another work area for a number of reasons. If he assigns a machine, it is unlikely that he would want to wait for the machine to finish before moving on to deal with tasks on other work areas. If no tasks exists on the current work area the operator will want to switch because of the probability that tasks exist elsewhere (unless the task class of the current work area has a high arrival rate and/or a high reward.).

There are other reasons an operator might not want to switch to another work area, even if the present one

contains nothing for him to do. The primary reason is that transitions between work areas take an exogenously specified transition time (TT). When the operator gives the command to switch work areas, the display of current work area goes blank for TT seconds. Over this interval, the operator can do nothing constructive. If the transition time were very long it might pay for the operator to wait for a new task appearance in his current work area instead of switching to some unknown one.

There are other reasons an the operator might not want to switch to other work areas. One such reason is that there are machines currently dealing with tasks on those work areas. Even if the other work areas possibly contain unattended tasks the operator might have a hunch that a task is about to appear where he is.

Essentially there are only two problems facing a human in a multi-work area environment. The first is to decide which work area he wants to be in. That is, he must decide whether to stay put or to switch to another work area. If he wants to switch, he must choose his destination. The second problem to be dealt with arises when a task is discovered. Should an operator do the task himself, or should he assign a machine to it thereby, freeing him to examine other work areas? How an operator faces these two problems will determine his performance in dealing with his environment.

5.3 Building the Normative Model of SUPER

It is necessary to find an optimal strategy for dealing with the game represented in SUPER in order to have some baseline with which to compare observed operator performance. Because the nature of the paradigm is so well defined it should be possible to generate such an optimal strategy based on the parameters presented.

In determining just which option is the "optimal" strategy, the assumption is made that the human subject will try to maximize his net reward (minus costs) over the course of a simulation. In the analysis operator performance was made independent of simulation time by considering expected reward per unit time as the dimension to be maximized. In an environment with an infinite time horizon, or one long enough that the operator cannot forsee an immediate end to his work, an operator will try to increase his absolute score as fast as possible: he will be trying to maximize his net reward per unit time.

In a situation where the operator has M machine aids and faces task arrivals in R work areas, the operator's set of possible actions can be reduced to a limited number. In

the general case he will have only R+M+1 options. He can do the task in the current work area himself (1 option); he can assign one of his machine aids (M options); he can stay in the current work area without doing a task (1 option); or he can make a transition to another work area (R-1 options). At any point in time, the operator's decision tree will have R+M+1 branches.

Naturally, in a specific situations, many of these branches will be inappropriate. For instance, if no t exists on the current work area then the operator will only have the options of waiting or transfering to another work area (R branches). In the case where all machines are identical, the operator will not have to choose between them. Assigning any machine becomes only one option. The number of options will be reduced to R+2.

Regardless of the number of options, the particular branch that is optimal will depend on the particular situation faced by the operator.

5.3.1 The Decision to Change Work Areas

Consider the decision most often faced by an operator in a multi-task/multi-work area environment: "When should I switch from the present work area to another? Should I do it now or later?" This is not a simple choice to make. The operator's action will depend upon the likelihood that tasks exist on other areas, the value of completing those tasks, the penalty for not doing them and the transition time required to reach those tasks.

In order to analyse the subtleties of the decision to change work areas, a simplified version of SUPER was considered in which there are only two work areas containing task classes with identical parameters. The use of machine aids was prohibited in this initial analysis.

In such a simple two-identical-work-area/no-machine environment it is easy to completely characterize the strategy employed by an operator. There are two components to this strategy:

> 1) If a task exists in the operator's current work area then he will complete it. This assumption logically follows from recognizing that an optimal strategy is required. In an optimal strategy the operator will never transfer, and therefore never find himself in a work area where he has no intention of doing tasks. The best the operator could hope for by switching

would be to find an identical task in the other work area.

2) If no task exists in the operator's current work area he will wait until the probability that a task exists elsewhere exceeds some critical probability, PC. He will then transfer to the second work area.

The operator's decision to switch work areas has been reduced to the single parameter PC which will depend on the characterics on the task environment summarized in Figure 5.3. Recall that tasks are the same in both work areas.

L : arrival rate for each work area (tasks/time)

u : service rate of tasks by operator (tasks/time)

h : holding cost for tasks (cost/time)

R : reward for completing a task (reward)

TT : transition time for movements between work areas (time)

Figure 5.3 : Task parameters for an environment with two work areas and no machine aids

The results obtained in an analysis of this two work area game can be generalized up to situations with many work areas. In any situation there are really only two classes of work areas: the one where the operator is and the ones where he is not. By making this distinction any situation represented by SUPER can be reduced to a two work area environment.

Additionally, it is likely that PC will take on values of either zero or one in most situations, corresponding, respectively, to situations where the operator will switch work areas immediately if there is nothing to do and to situations where he will not switch at all. However, in some instances PC will be between these extreme values, and the operator not want to change work areas unless he is fairly certain he will find a task needing attention at his destination.

In the two-work-area/no-machine game, choosing the optimal operator strategy amounts to finding the value of PC

which miminizes the operator's expected reward per unit time. By definition, if the probability that a task exists on the other work area exceeds PC the operator will switch to the other work area. The probability that a task exists on a given work area, where no task existed t seconds earlier, can be computed by knowing the distribution of arrival times. This distribution is assumed to be exponential with parameter L, the mean arrival time (see Appendix A). An exponential distribution has the following form:

$$f(t)=EX(t,L)=L*\exp(-L*t)$$

The probability a task exists on another work area, PO, is simply the probability that the task arrival time is less than T.

$$PO(T)=\int_0^T EX(t,L)dt$$
$$= 1 - \exp(-L*T)$$

By setting PO equal to PC we can determine the time an operator will wait before returning to a work area he had previously left. Humans think better in terms of "time" than in terms of "probability". By setting a critical probability, PC,we are really specifying a critical waiting time, tw, for the operator.

$$PO(tw)=PC$$
$$1-\exp(-L*tw)=PC$$
$$tw=-\log(1-PC)/L$$

For PC equal to one, tw is infinite and the operator will never transfer from his current work area. The operator will wait, on average, 1/L seconds for tasks to appear. He will do these tasks, on average, in 1/u seconds for which he will recieve reward R minus the net holding cost h/u. The cycle will then start again. Ultimately, a task will probably appear on the work area the operator chooses to ignore, which will exercise the holding cost h. In this steady state situation the net expected reward per unit time, RPT, can be computed.

$$RPT(PC=1) = \frac{R-h/u}{1/u + 1/L} - h$$

Unfortunately, RPT is not so easily calculated for values of PC less than one. When the operator does switch work areas, the analysis becomes more complex. One method

for dealing with this complexity is to analyse the system using Markov decision theory (Howard [12]).

In Markov decision theory a probabilistic system is modeled as being in one of several well-defined states. When the system is in one of these states it will make a transition into other states with a specified probability for each transition. Each transition also has an assosciated reward. The transition probabilities and rewards will vary according to the strategy used by the system controller.

5.3.2 The States in a Two-Identical-Work-Area / No-Machine Scenario

By looking at the two-work-area/no-machine game at specific "time windows" corresponding to immediately before the operator begins work on a task or immediately after a task is completed, the system can be characterized as having only four distinct states. At the times of interest the system will always be in one of the following states:

State 1 : no tasks exist

State 2 : one task exists and it is in the work area currently being scanned by the operator

State 3 : one task exists but not in the same work area as the operator

State 4 : two tasks exist, one in each work area

Because the two work areas contain tasks with identical parameters the state definitions do not have to distinguish between the different work areas. All that is important is whether a solitary task exists in the same work area as the operater (termed the "current work area"), or in the other work area.

It is important to note if no task exists in the current work area the operator cannot tell if the system is in State 1 or State 3. The operator only has certain knowledge about the current work area, though he can make educated guesses about task existence on the other. Similarly he cannot distinguish between State 2 and State 4. The operator only knows if a task exists in the current work area. He knows how long ago he last visited the other work area and can compute the probability that a task has appeared there. If the current work area is empty (State 1 or State 3) the operator will wait until that probability

exceed PC, then he will change work areas. If a task does exist on the current work area the operator will do it.

Figure 5.4 shows a state diagram of the system complete with all possible state transitions. T(i,j) denotes a possible transition from State i to State j.

Consider all the transition from State 1. The operator will be wait in the current work area until PC is exceeded, then he will switch. At any time tasks can appear in either of the two work areas. If a task appears in the current work area while he is waiting there, the system will enter State 2 if nothing has happened on the other work area. If a task has appeared there, however, the system will have entered State 4. If nothing happens in the current work area while the operator is waiting there the system might still enter State 3, without the operator's knowledge, if a task appears on other work area.

If the operator decides to leave an unoccupied work area while the system is in State 1 four possibilities might occur while he is in transit. A task might appear on his destination; a task might appear at his point of origin; tasks might appear in both work areas; or no task appear. When the operator emerges from his transition, the system will have transferred to State 2, State 3, State 4 or State 1 respectively.

Suppose the system is in State 2, with a single task on the current work area. The operator will do this task. If a task occurs on the other work area, the system will be in State 3 when the operator finishes his task. If not, the system will be in State 1.

From State 3, the system can go to only States 2 or 4. The operator will wait in the unoccupied state and then transfer to the other one. If a task appears on the current work area at any point over this time interval the system will be in State 4. Otherwise it will be in State 2.

If the system is in State 4 then tasks exist on both work areas. The operator will do the task in the current work area placing the system immediately in State 3 where only one task exists and it is in the other work area.

5.3.3 State Probabilities

Each of the transitions in Figure 5.4 is associated with a state transition probability matrix P(i,j). P(i,j) is the probability that a transition will occur from State i to State j, given that the system started in State i. By definition, the sum of P(i,j) over all possible destination states j is 1.0. As an example, P(4,3) is one because the

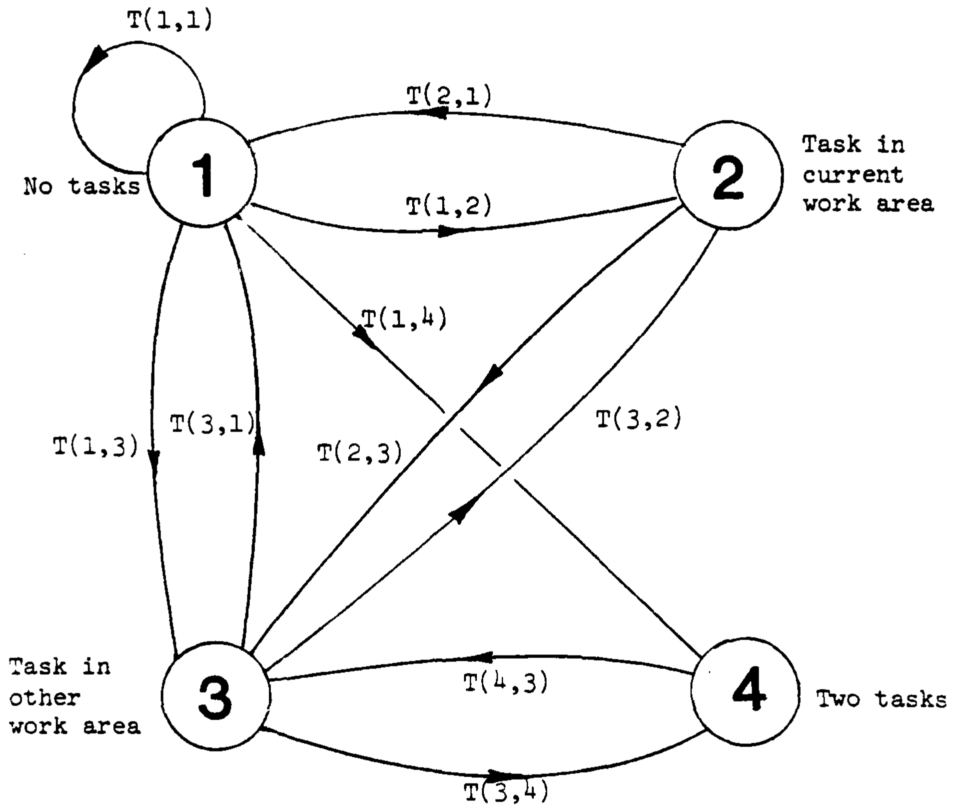

Figure 5.4: State transition diagram for the task environment with two identical work areas and no machine aids

system can only enter State 3 when it is in State 4. Similarly, P(2,3) equals the probability that a task arrives in the other work area while the task in the current work area is being dealt with. Since arrival times are exponentially distributed, P(2,3) can be calculated as follows (see Appendix A).

$$P(2,3)=PO(1/u,L)$$
$$P(2,3)=1-\exp(-L/u)$$

and

$$P(2,1)=1-P(2,3)$$
$$P(2,1)=\exp(-L/u)$$

The value L is the mean arrival rate [arrivals/second] for the tasks and u is the mean service time [1/service time]. The other members of the state transition matrix P(i,j) can be calculated in a similar manner.

Consider a very long sequence of state transitions. A certain fraction of these transitions will originate in each of the four system states. In the long run this fraction will approach a steady state value called the State Probability, PI(i). Howard [12] gives the following formula from which PI can be derived.

$$PI=PI*P$$

5.3.4 Expected Reward/Time of an Operator's Strategy

PI(i) gives the fraction of transitions that originate in State i. This is different from the fraction of time spent in each state because the time spent in each state varies from state to state. The time spent in States 2 and 4 will be equal to the service time of a task (1/u). The time spent in States 1 and 3, however, will depend on the operator's strategy (parameterized by the critical probability PC) and the arrival rates for tasks.

The expected time spent in each state can be multiplied by the state probabilities to determine the average time between transitions.

Similarly, the average reward of each transition can be computed by multiplying the expected reward for each transition from State i by the state probabilities, PI(i). The expected reward for transitions from State i are computed as the sum over all destinations, j, of P(i,j)xR(i,j) where R(i,j) is the expected reward for a transition from State i to State j.

R(4,2) equals the reward for completing a task minus

the holding cost for that task and for the one in the other work area while the system was in State 4.

$$R(4,2)=R-2h/u$$

Similarly, the expected reward of a transition from State 2 to State 3 includes the net reward for completing a task (R-h/u) minus the expected cost for the new task occurance on the other work area (see Equation A.4 in Appendix A for a derivation of this cost).

5.3.5 Determining an Optimal Strategy

As described above, an operator strategy can be completely described by the critical probability, PC. Many of the transition times and costs will depend on the value of PC. The expected reward per unit time for the system can be computed as a function of PC. For any set of task parameters, reward per time (RPT) can be optimized over PC in order to generate the best value for PC.

There are three independent quantities that describe a set of task parameters. These are the ratio of the reward to the holding cost [RRTH=R/h], the ratio of the service rate to the arrival rate [RUTL=u/L] and the ratio of the transition time to the mean arrival time [PTTL=TT/(1/L)=TT*L]. The best strategy for two cases with the same values for these three numbers will be the same regardless of the absolute value of the task parameters. Figure 5.5 lists the optimal value for PC as a function of these three parameters.

In general, the best value for PC is either zero, corresponding to switching work areas immediately, and one which corresponds to a policy of never changing work areas. When transition time (TT) increase PC goes to one because so much time would be spent in transition that it would not pay to switch. For small arrival rates (increasing PTTL and RUTL) PC also increases because as interarrival times grow, an operator will have a greater likelihood of finding a task by looking elsewhere than by waiting for one to appear.

RATIO OF REWARD TO HOLDING COST = 0.0

PTTL= \ RUTL=	1.00	2.00	2.50	3.00	4.00
0.00	0.000	0.000	0.000	0.000	0.000
0.25	0.000	0.000	0.000	0.250	0.250
0.50	0.450	0.650	1.000	1.000	1.000
0.75	1.000	1.000	1.000	1.000	1.000
1.00	1.000	1.000	1.000	1.000	1.000
1.50	1.000	1.000	1.000	1.000	1.000
2.00	1.000	1.000	1.000	1.000	1.000

RATIO OF REWARD TO HOLDING COST = 10.0

PTTL= \ RUTL=	1.00	2.00	2.50	3.00	4.00
0.00	0.000	0.000	0.000	0.000	0.000
0.25	0.000	0.000	0.000	0.000	0.000
0.50	0.450	0.700	0.750	0.750	0.800
0.75	0.900	0.900	0.900	0.900	0.950
1.00	1.000	1.000	1.000	1.000	1.000
1.50	1.000	1.000	1.000	1.000	1.000
2.00	1.000	1.000	1.000	1.000	1.000

RATIO OF REWARD TO HOLDING COST = 50.0

PTTL= \ RUTL=	1.00	2.00	2.50	3.00	4.00
0.00	0.000	0.000	0.000	0.000	0.000
0.25	0.000	0.000	0.000	0.000	0.000
0.50	0.000	0.750	0.750	0.750	0.750
0.75	0.850	0.900	0.900	0.900	0.900
1.00	0.950	0.950	0.950	0.950	0.950
1.50	1.000	1.000	1.000	1.000	1.000
2.00	1.000	1.000	1.000	1.000	1.000

RATIO OF REWARD TO HOLDING COST = INFINITE (i.e. Holding cost = 0)

PTTL= \ RUTL=	1.00	2.00	2.50	3.00	4.00
0.00	0.000	0.000	0.000	0.000	0.000
0.25	0.000	0.000	0.000	0.000	0.000
0.50	0.700	0.750	0.750	0.750	0.750
0.75	0.850	0.850	0.850	0.850	0.850
1.00	0.900	0.950	0.950	0.950	0.950
1.50	1.000	1.000	1.000	1.000	1.000
2.00	1.000	1.000	1.000	1.000	1.000

Figure 5.5: The optimal value of the critical transition probability PC calculated in the two-work-area / no-machine Markov model

CHAPTER 6

UTILITY IN AN ENVIRONMENT OF CERTAINTY

6.1 The Linear Utility Assumption

The assumption that subjects have a linear utility for the rewards in the experimental paradigm used in this study is basic to the contention that the models developed for operator performance reflect optimal behavior. Sub-optimal performance on the part of an experimental subject should reflect some psychological barrier that is preventing the operator from fully grasping the complexities of the task environment. However, it may also be the case that the reward function the operator is trying to maximize is not the same one as the one the model maximizes. The subject may be acting optimally based on his internal value system.

Suppose the human operator is found making decisions in a multi-task environment that the model shows to be suboptimal. One possible explanation for this sub-optimality is that the human is incorrectly perceiving the probabilities that tasks exist in other work areas. Alternately, he might just be overcome with the choices available to him. It is also possible that the human simply does not value the rewards and costs presented in the game at their face value; that is, the operator's "utility" for points is not linearly related to the physical "magnitude" of the points.

There are certain reasons for making the assumption of linear utility. First, the point system used in these games has no correlation to physical quantities, such as money. Game rewards cannot be interpreted as real-world rewards and it is therfore unlikely that humans will have any particular preferences for different values. As a result, the utility for points will be linearly related to the points themselves.

Secondly, the experimental situation is only a game. In real life most people tend to be risk adverse in the search for rewards. In a gaming situation where there score can be compared to that of other subjects there is no reason to be risk adverse. A bigger danger is that over-competitive subjects will take risks to improve their score and outperform other subjects.

Despite these arguments, the linear utility assumption cannot be blindly accepted. Even with no physical meaning to the scores humans may not act in a linear fashion. Consider a simple experiment where two numbers are presented to a subject and he is asked to determine if they are equal or not. The response time for the pair 3-5 is less than for the pair 7-9. Though the absolute difference is the same in both cases, the "perceptual difference" apparently is not. This result would violate the linear utility assumption.

6.2 Testing the Linear Utility Assumption

In order to test the hypothesis that linear utility was an acceptable assumption an alternate game was considered. This game was similar to SUPER in that an operator had to deal with tasks by either doing them himself or by assigning them to a costly machine aid. The game differed from SUPER in that no new tasks appeared over the course of the experiment and all tasks were visible at all times. By having operator duties similar to those in SUPER but in a very simple environment, and by removing the source of uncertainty, any sub-optimal behavior could be attributed to misconceptions about the value of rewards and costs.

The playing field for the game is shown in Figure 6.1. The name of the game, BOXCLR, derives from the fact that the operator was faced with a number of boxes, or tasks, that had to be cleared from the screen. All tasks had the same reward, R, for completion and the holding cost, h [points/second], for being left unattended. The operator could "do" a task himself by placing a cursor in the extreme left column on the same row as the desired task and depressing a "DO TASK" button on a control box.

The column was called a garage, and a little bulldozer would leave the garage, move out to the task, turn around, and then push the task/box back to the garage where both would leave the playing field.

Alternately, the operator could place the cursor in a garage/column towards the center of the screen, and press the "ASSIGN MACHINE" button. This action would assign another bulldozer leave this second garage and get the box. This bulldozer represented a machine aid. A wage, w, had to be paid for use of the machine aid [points/second].

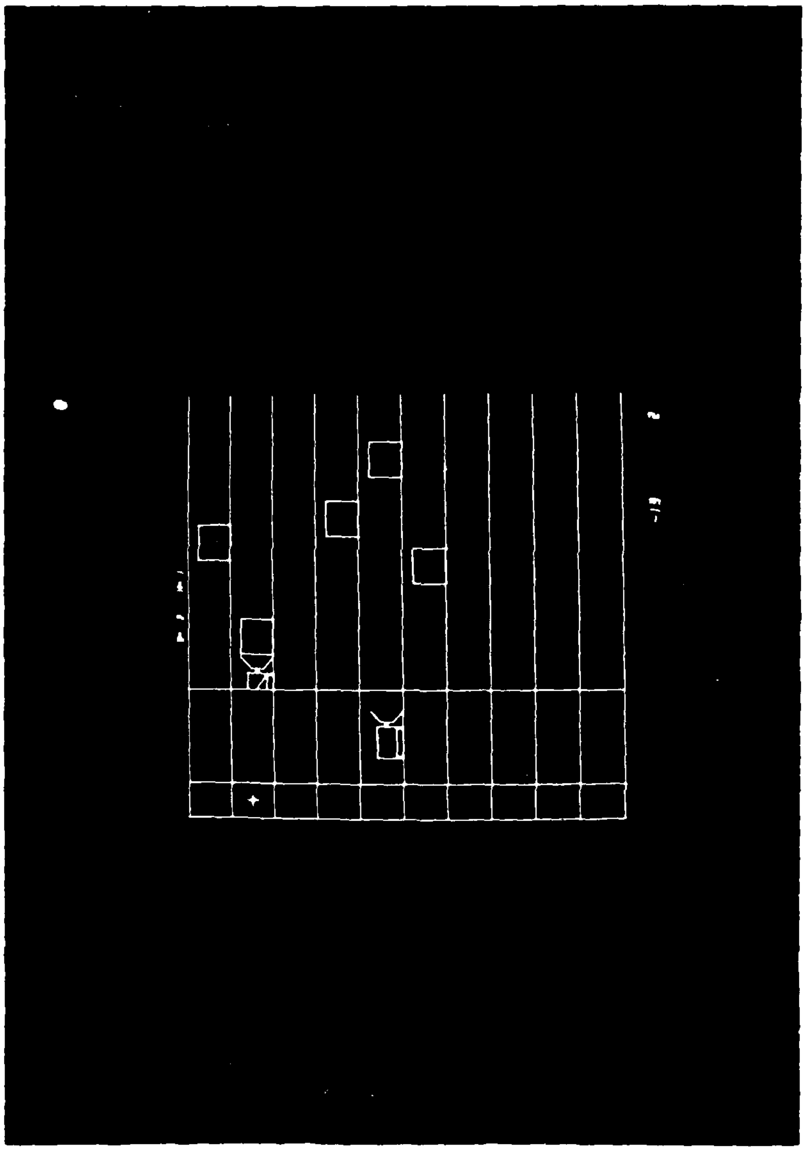

Figure 6.1(a): Photograph of the BOXCLR playing field

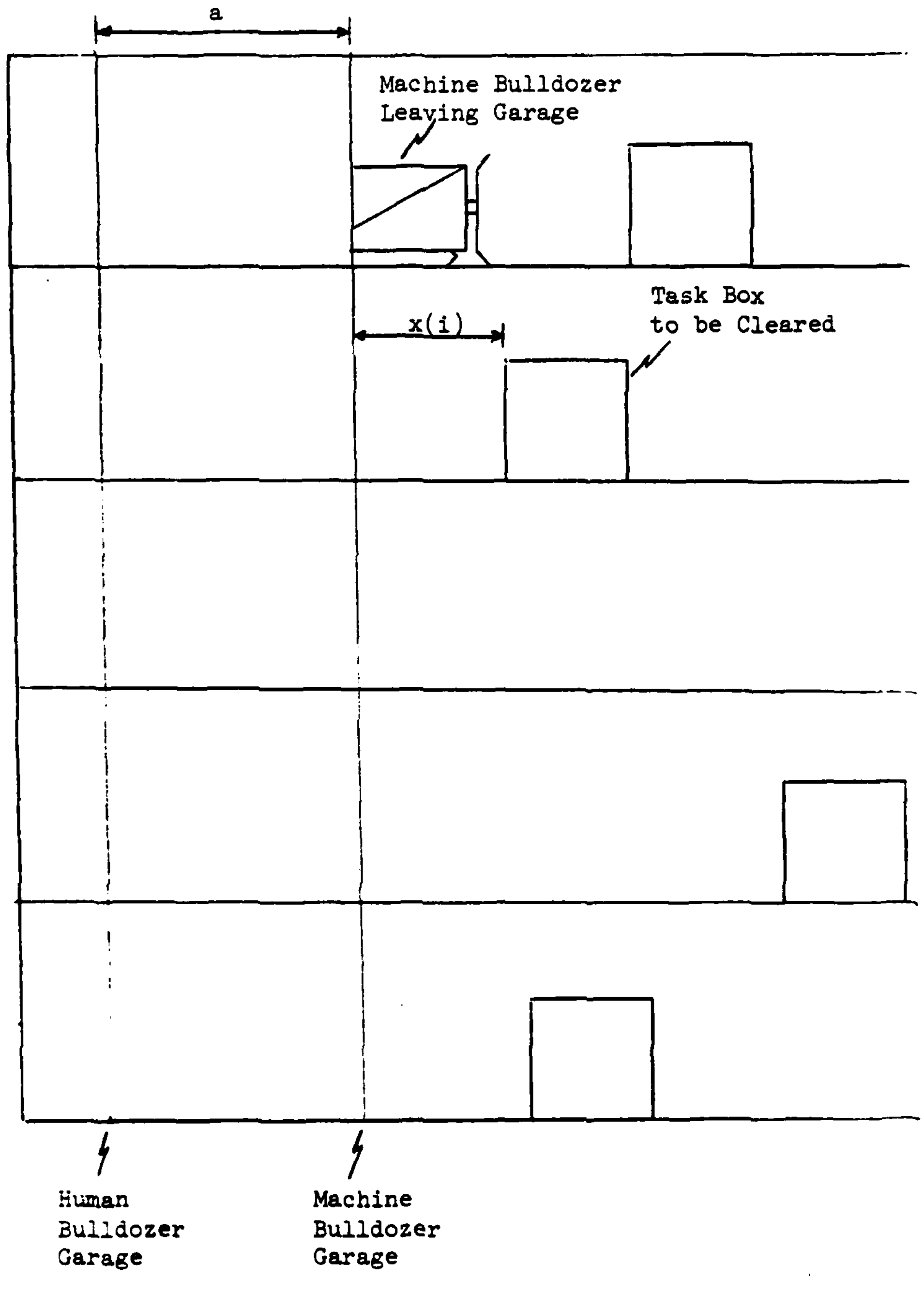

Figure 6.1(b): Diagram of the BOXCLR playing field showing parameters x(i) and a used in analysis

This machine bulldozer completed tasks faster than the human could. Though both bulldozers moved at the same speed, the second one left from, and returned to a garage that is closer to the box. The cheap bulldozer, in the leftmost garage, correponded to the human operator doing the task himself.

The operator had to be pressing either the DO TASK or the ASSIGN MACHINE button in order for either bulldozer to move. When neither button was depressed, all the action stopped: the clock stopped, holding costs and machine wages were not paid and neither bulldozer moved. This time out allowed the operator to plan his future strategy without worrying about the ticking clock. If the operator acted sub-optimally it could not be attributed to rushed decisions.

The operator's chief concern was to assign the bulldozers to the various tasks based on the relative rewards, holding costs and machine wages. Only one bulldozer could leave each garage at any time. When all the boxes had been cleared from the screen, then the experiment was over.

6.3 Measuring Operator Performance

BOXCLR is a very straightforward game. The operator is faced with N tasks which can each be dealt with in two ways: by machine or by human. An operator strategy is determined by which set of tasks he chooses to do by hand and which set he assigns to a machine.

Given two tasks in the same set, an operator should do the one closest to a garage first. The closer a task is to a garage, the less time it will require before it is completed. The order the operator chooses to do these two tasks will not affect the total time for service, which will equal the sum of the individual service times. However, the order will affect the total net holding cost incurred for both tasks. Both tasks will exist while the first one is being dealt with. Only one task will exist over the service time of the second task done. Total holding cost will therefore be minimized by doing the shortest task first.

The value of the reward, R, obtained by completing each task should have no effect on operator strategy. Regardless of ordering, the operator will receive R points for each of the N tasks. Only the cost variables, holding cost and machine wage, will cause different strategies to yield different scores.

Because each of the N tasks can be handled one of two ways, by hand or by machine, there are 2**N possible

strategies (assuming that of the tasks done by hand, the shortest are done first, and that of the tasks done by machine the shortest are again done first). The total cost of a strategy is simply the sum of the cost for those tasks done by hand, and those tasks done by machine. The total cost of those tasks done by hand is equal to:

$$\text{HUMAN COST} = \sum_{i} \frac{2*[x(i) + a]}{spd} * (M-i+1)*h$$

where

x(i) : the position of task boxes done by hand [see Figure 6.1(b)]

a : the difference in between the human garage and the machine garage [see Figure 6.1(b)]

h : holding cost (units/time)

spd : the bulldozer speed (distance/time)

M : the number of tasks by hand

i : the index of hand-done tasks in order of service time [i.e. x(i) < x(k) for i < k] and i=1,M

This formula assumes the shortest task done by hand is undertaken first. While it is being dealt with all other hand-done tasks must wait and incur the holding cost. When the ith task is being done all longer tasks must similarly wait for service. The factor "2" arises from the fact that bulldozers must first move to the box from the garage as well as push the box back to the garage. The cost of all tasks done by machine is based on a similar formula with the machine wage included.

$$\text{MACHINE COST} = \sum_{j} \frac{2*x(j)}{spd} * (N-M-j+1)*h + w$$

where

x(j) : the position of task boxes (see Fig. 6.1) done by machine

w : the mahcine wage (cost units/time)

N : the number of tasks

N-M : the number of tasks by machine

j : the index of machine-done tasks in order of service time [i.e. $x(j) < x(k)$ for $j < k$] and $j=1,N-M$

With these two formula it is possible to compute the total cost of any possible operator strategy. The program TCST, shown in Appendix B does just this. An additional routine, HSTGRM, determines each of the 2**N possible sequences, computes their cost using TCST, finds the optimal strategy with the lowest total cost and generates a histogram showing the number of strategies that yield particular total costs. One such histogram is shown in Figure 6.2. It is possible for there to be more than one "optimal" strategy in a simple game like BOXCLR because two or more divisions of tasks between man and machine can yeild the same overall cost.

When an experimental subject attacks the game in BOXCLR the cost of his strategy can be computed by seeing which tasks he does by hand and which he does by machine aid. Using TCST the cost of this strategy is computed and compared to the output from HSTGRM. This process yields the fraction of all possible strategies that would have been better than the one actually used, and the cost of the subject's strategy as a fraction of the minimum cost possible.

The cost of operator strategy used for comparison with the output of HSTGRM was not the actual cost incurred by the operator when he applied his strategy. Human subjects tend to waste time as they make small mistakes in applying their chosen strategy. Because of implementation error, an experimental subject applying the optimal strategy will generate a sub-optimal score. As a result, a subject's score is not directly comparable to the output from HSTGRM which assumes that an operator can carry out a desired strategy perfectly.

However, in BOXCLR the concern is whether or not a subject can determine which is his best course of action, not whether he can carry out his plans flawlessly. By looking only at the subject's plans, and not his execution of those plans, it is possible to see if he is indeed acting close to optimal. If he is at least trying to attain the optimal behavior described in TCST, which assumes a linear utility for cost, then it can be concluded that the human subject has linear utility for the point system used.

For each experiment, the subject's strategy was recorded. Using TCST, the maximum potential score from applying this specific strategy could be calculated.

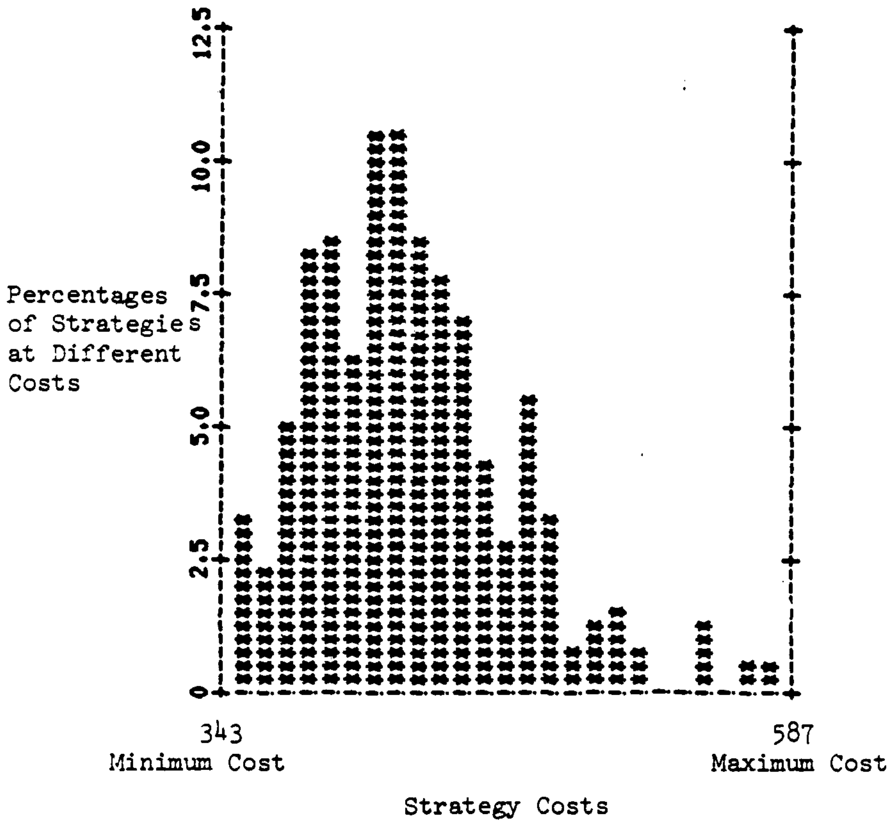

Figure 6.2: Sample histogram showing the distribution of strategy costs in BOXCLR for the case of eight boxes and a machine wage of four points per second

Two measures of operator performance were considered. The first of these were the fraction of all possible strategies for a particular experiment that were better than the one employed by the operator.

The second measure takes into account the dispersion of the histogram of all possible strategies. If many strategies are very good the operator may have achieved close to the minimum cost, but still have a large fraction of strategies better than his. The second performance measure is the operator's realized costs divided by the minimum possible cost.

6.4 Experimentation

Three subjects each conducted 96 trials with BOXCLR. The holding cost was fixed at one point per second while the machine wage was either zero, one, two or four points per second. In addition the number of boxes that needed to be cleared changed from two to four to eight. The order of the experiments was random.

Table 6.1 and 6.2 shows average operator performance as a function of the two experimental parameters.

		Machine Wage (points/second) 0	1	2	4	Average
No.	2	.005	.047	.056	.099	.052
of	4	.030	.093	.045	.033	.050
Boxes	8	.037	.050	.038	.152	.069
Average		.024	.063	.046	.095	.057

Table 6.1 : Fraction of all strategies better than the one used by subjects

		Machine Wage (points/second) 0	1	2	4	Average
No.	2	1.003	1.022	1.034	1.078	1.034
of	4	1.008	1.042	1.030	1.039	1.030
Boxes	8	1.020	1.026	1.031	1.091	1.042
Average		1.010	1.030	1.031	1.069	1.035

Table 6.2 : The ratio of operator Cost to minimum cost for experimental subjects

Tables 6.1 and 6.2 show that human subjects do tend to be close to the optimal strategy. Because this optimal strategy is based on the assumption of linear utility, then a safe assumption would be that human subjects do have linear utility for the points used in the game.

The similarities between BOXCLR and SUPER are numerous. Based on these results the linear utility assumption used in the analysis of SUPER is valid.

There is a slight decline in performance as, N, the number of boxes increases. This is because complexity of the task of choosing the best strategies grows with N. In fact, the number of possible strategies is 2**N. The implication of this is that subject performance drops with complexity of the decision algorithm, though for simple enough situations they act as if they do have linear utility.

The decline in performance with increasing wage shown in Tables 6.1 and 6.2 might imply that subjects don't have perfect linear utility. This trend is due to the fact that the problem seems more complex to subjects as wage increases. The magnitude of the trend is heavily weighted by the very poor showing of subjects for a wage of four points per second with eight boxes.

CHAPTER 7

THE OPTIMAL DECISION MODEL

7.1 Limitations of the Markov Model

The Markov model approach is easy to apply in the simple case of no machine and two identical work areas where only four states are needed to describe the system. When a machine aid is added and the number of work areas increases the number of distinct states required to describe the system increases.

The simple model given in Chapter 5 has only four states corresponding to: no tasks; one task in the current work area; one task in the other work area; and two tasks. If the tasks in the two work areas are no longer identical, however, a complete description of the state of the system requires information on which work area is occupied by the operator. As a result, twice as many states are necessary to characterize the system.

The problem becomes even more complex when the number of work areas is greater than two. When there are only two work areas it is easy to see that upon arrival in one work area the operator is departing the other one. With more work areas, arrivals in one work area say nothing about when the operator was last in any other work area. This information concerning the elapsed time since each work area has last been examined is important to the operator because the greater the elapsed time the greater the probability that a task arrival has occured.

In order to completely and unambiguously describe the state of the operator's environment, both the physical state of the system (i.e. where tasks exists) and the operator's information about that state must be considered. The operator's actions will depend upon his information and the

effects of these actions will depend on the physical state of the system. Modeling the system must take both into account.

Consider a case with R work areas. Each work area will either have a task on it or it won't. Just representing the possible combinations for task existence will require 2**R states. Additionally, the state will also depend on the location of the machine aids. There are R+1 possible locations for each machine corresponding to the R work areas and to the condition that the machine is not assigned anywhere.

The operator will similarly have R+1 possible locations. Specifying the locations of the operator will require increasing the number of states by a factor of R+1. The same increase will be necessary to specify the location on each machine. The net increase in states will be (R+1)**(M+1).

An optimal policy based on a Markov analysis would generate a list of the best next move for each state. However, such an analysis is useless if the operator does not know which state the system is in. Because the operator can only look at one work area at a time, his information about vaious work areas is outdated. The strategy chosen by an operator will depend on his knowledge of the system. However, the effects of this strategy will depend on the actual state of the system. In modeling the system, both the actual state of the system and the operator's knowledge of that state must be represented.

If an operator has not visited a work area in a long time he may assume a task has appeared in his absence and act accordingly. In most cases his assumption will be correct but in an optimal analysis the possibility that no task has appeared must also be considered. For each level of knowledge, every possible system state must be considered.

The operator's state of knowledge of the system can be summarized as the time elapsed since he last examined each work area. An equivalent measure would be the probability that tasks exist on various work areas. Both these measures are continous and have an infinity of possible values. As an approximation they might be considered in L discrete levels. In this case, L**R states are needed to define the operators knowledge of the system. To record whether tasks existed would require 2**R states. The total number of states therefore becomes

$$\text{Required states} = 2^R (R+1)^{M+1} L^R$$

$$= (2L)^R (R+1)^{M+1}$$

The following table lists the required number of states as a function of the number of work areas (R) and the number of discrete categories for elapsed time (L). The simple case of no machine aids is considered here. Recall that for an exact model, elapsed time since last look must be continuous (i.e. L must be infinite).

		L divisions in elapsed time				
		1	2	5	10	100
R	2	12	48	300	1200	1.2×10^{4}
	4	80	1280	5×10^{4}	8×10^{5}	8×10^{9}
	6	448	28672	7×10^{6}	4.5×10^{8}	4.5×10^{14}
	8	2304	5.9×10^{5}	9×10^{8}	2.3×10^{11}	2.3×10^{19}
	10	11264	1.2×10^{7}	1.1×10^{11}	1.1×10^{14}	1.1×10^{24}

Table 7.1:- Required states for representing the R work area, L time division, no machine environment

The above table shows that it becomes impractical to generate a model similar to that in Chapter 5 when R increases and when the tasks are not identical because of the high number of required states. Fortunately, approximations can be made that make the problem more tractable.

The Markov model approach generates an analytic solution for the expected reward per unit of time of a given strategy. This expected reward per time can bè approximated by employing the specific strategy in a multi-task situation and recording the actual rewards and times. The longer the time this simulation is employed, the closer the observed reward per time (RPT) will approach the exact RPT. Figure 7.1 gives a sample time series showing RPT for a two-work-area/no-machine game at various time intervals. The operator in the simulation is assumed to change work areas whenever he finds himself in a work area not containing a task to do. The exact RPT, as computed with

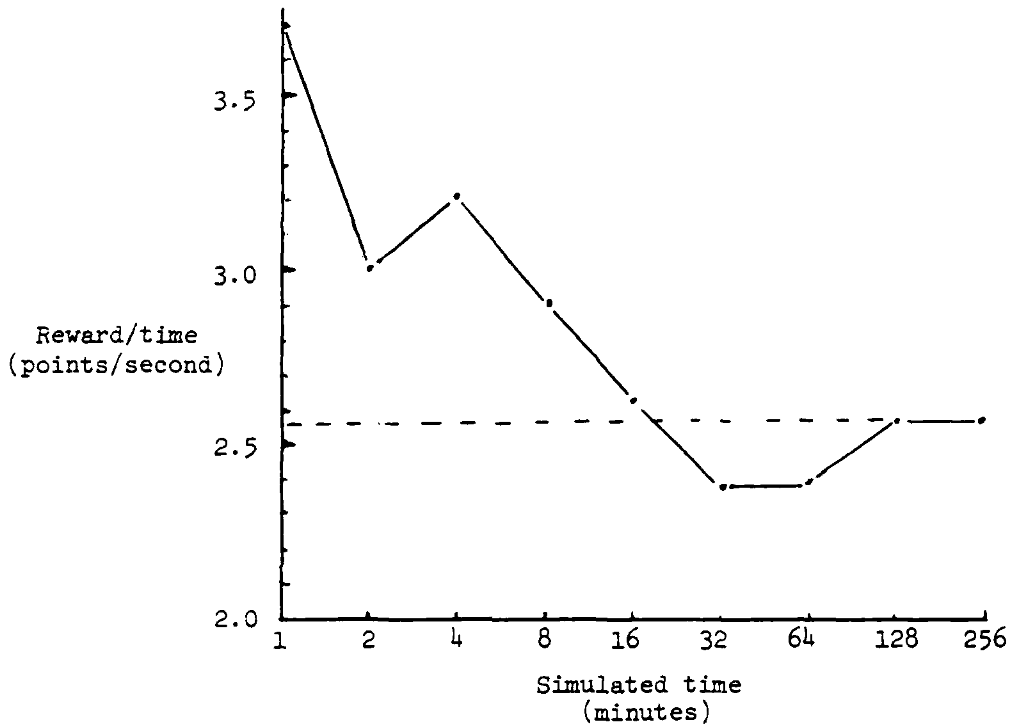

Figure 7.1: The reward per time observed at various points in a simulation of SUPER. The dotted line gives the analytically calculated long run reward per time.

the queuing theory model presented in Chapter 5, is also shown.

7.2 The Simulation, or Monte Carlo, Approach

If the strategy is well defined, the computer can be asked to play the part of an experimental subject. The computer can implement the desired strategy far more consistently than can a human. To speed computation the the graphic display used by the human does not have to be drawn for the computer. Similarly, the computer not need to play the game in real time. By using the computer in a stripped-down/speeded-up version of SUPER, a hundred seconds of real time playing can be simulated in only seven seconds.

This simulation approach can be used to generate estimates for the RPT of any strategy that can be "taught" to the computer. Naturally, if a strategy requires extensive computation in determining which course of action to take at various points in time, the time spent on the simulation will increase. The estimation technique whereby a reward function for a probabilistic system is computed through simulation is commonly called a Monte Carlo approach.

The chief question raised in this simulation approach is just *exactly how long should the simulation* be run before the estimate is a sufficient representation of the actual RPT. Figure 7.1 shows values of RPT that vary significantly before converging.

In this analysis, the computer was asked to check its estimate of RPT after each simulated 100 seconds. The estimate was compared to the previous estimate obtained 100 seconds previously. If the absolute difference between the two estimates was less than a critical fraction of the latest value of RPT then the simulation ended and the estimate of RPT was recorded. This critical fraction was called the Simulation Confidence Factor, or SCF. The SCF is a measure of the convergence of the simulation and of the accuracy in the estimate of RPT.

Table 7.2 gives the time required to achieve various levels of accuracy using this method. The actual time required to run to longest of these simulations on the PDP 11/34 computer at the Man-Machine Systems Laboratory at M.I.T. was about ten minutes.

SCF	Simulated Time (secs)
.1	200
.01	1000
.001	1400
.0001	8000
.00001	13400

Simulation stops when CHANGE IN RPT < SCF*RPT

Table 7.2: Time required to estimate RPT to various levelsof accuracy

The accuracy results presented in Table 7.2 apply to experimental subjects as well as computer simulations. The accuracy in the estimation of RPT increase with the length of simulation. To get comparable levels of accuracy for the RPT of experimental subjects the length of experiments must be on the same order as the simulated times in Table 7.2. In order to have an accuracy to a level of one-in-a-thousand will require a simulation of 8000 seconds. This corresponds to just over two hours of continuous experimentation with one particular subject and one particular set of parameters.

Alternate performance measures are of course possible and will be discussed later. One such measure involves comparing the actual decisions made by an operator to a set of optimal decisions. The fraction of agreements would provide a measure of subject performance relative to the optimal.

7.3 Using a Decision Tree to Generate Optimal Stategies

In a situation where all task classes are identical, where the machine wage is zero, and where the man and the machine aid are interchangeable, a poosible canditate for the optimal strategy is one where the operator always switches to the work area where the probability of task occurance is greatest. If a task exists he should assign the machine to it. If the machine is unavailable, then he should work on the task himself until the machine is free. Using the simulation approach outlined above it is possible to compute an estimate of the expected reward per time of such a strategy. Other possible strategies can also be tested and the best of this set discovered.

This hunt-and-trial method will only uncover an optimal strategy if the optimal strategy is among the set hypothesized. For simple situations, such as the one mentioned above, it is not too difficult to guess which strategies will be effective. However, for complex scenarios with differing task parameters and multiple

machine aids the options are not so clear cut.

Consider the general case of an operator and M machine aids in an R task environment where the parameters of the individual task classes differ according to service time, arrival rate, rewards and costs, and where the productivity of the machine aids varies by task class. The decision to assign a machine to a particular task will depend on the likelihood that other tasks exist elsewhere and on the status of the other machines (i.e. is a more appropriate machine that is presently occupied about to become free ?). If the operator decides to change work areas, his destination will be chosen on the basis of task rewards and costs as well as the probability of task occurances in the other work areas.

It is impossible to use the method shown in Chapter 5 because of the vast number of states that would be required to specify the system. However, this problem does not mean decision theory must be abandoned. Instead, a traditional decision tree can be constructed which has a limited time horizon. The Markovian approach determined the best strategy taking all possible future events into consideration using, in effect, an infinite planning horizon. However, because events far in the future will have only a minor impact on a decision in the present, the infinite time horizon criterion can be discarded in order to make the decision problem tractable.

At any point in time the operator must decide to follow a course of action that is included in the following set:

1) He can do the task on hand by himself
2) He can assign a machine to do that task
3) He can transfer to another work area
4) He can do nothing (i.e. stay in the current work area)

Options 3 and 4 are very similar because staying in one work area and doing nothing is like transfering from that work area to itself. The operator's choice boils down to dealing with a task or moving to a work area. The operator's options are shown graphically in Figure 7.2 as a standard decision tree.

The decision to follow one of these options depends, in the general case, on the following variables:

- the elapsed time since the operator last looked at each work area, from which he estimates the probability that a task exists there

- the work area currently occupied by the operator

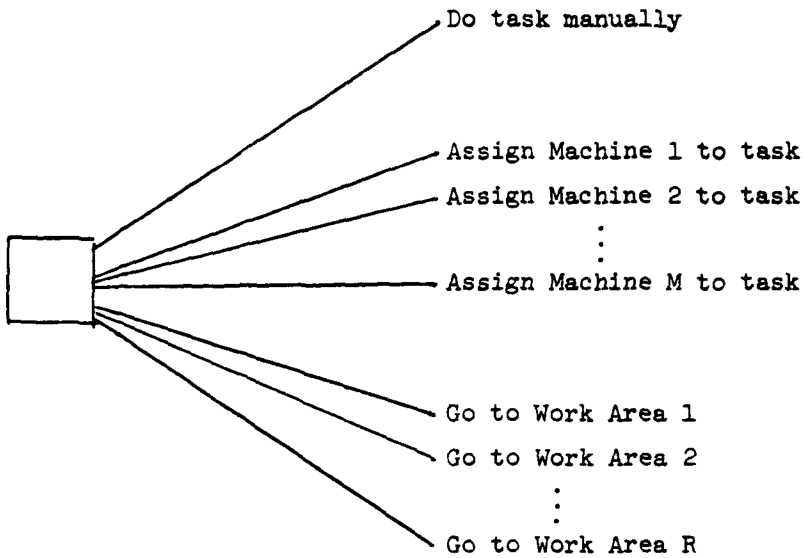

Figure 7.2: Options for an operator at each node in a decision tree, for a paradigm environment with M machine aids and R work areas

- the existence of tasks on the various work areas, if known by the operator

- the time until each machine aid will be available, assuming that it is currently occupied

- the game parameters [service time, arrival rate, holding cost, machine wage, and rewards]. If some tasks have been partially attended to, the operator must also know the fraction of the total service time that has elapsed.

These variables will completely describe the operator's potential state of knowledge about the man-machine system environment. When two situations arise having the same values for the above variables then a consistent operator will decide on the same course of action in both situations.

Each course of action shown in Figure 7.2 will commit the operator for a certain length of time. The decision to transfer to another work area will prevent the operator from doing anything else over the transition time. Assigning the machine, however, does not tie up the operator at all because assignment is instantaneous and the operator can immediately do something else.

The decision to work on a task by himself commits the operator to the service time of the task only when he wishes to complete the task. The operator can reevaluate his decision at any time. He may start working on a task only because the machine he wishes to assign is occupied elsewhere knowing that when the machine is free he will immediately reassign it to the task he is currently working on. The machine will then continue where the operator left off.

Similarly, the choice to remain in the current work area, even if no task is present can always be reevaluated. The operator may be just waiting for a task to appear. If a task exists, he may be waiting for a machine aid to become available.

So far, only the immediate consequences of the operator's decisions have been considered. In deciding on a particular course of action, the operator is limiting his future choices. In the case of only one machine aid, assigning the machine to a task ties up that machine for a length of time, and usually commits the operator to changing work areas. Once he has changed work areas he may discover another task which must be dealt with manually.

Alternatively, only empty work areas may be found and the operator will have to transfer again.

If the operator looks not only at his present decision, but at the future ones that subsequently will have to be made, he realizes that any decision in the present has far reaching consequences.

Figure 7.3 shows the expanded decision tree faced by the operator at for an operator looking three moves ahead. Only two work areas and one machine were considered in constructing this tree. The operator was assumed to start in a work area that does not contain a task.

Each path in Figure 7.3 over the planning horizon goes through three decision nodes where the operator must decide which branch to take. Additionally, all decisions to change work areas involve a chance node, which corresponds to the possibility that tasks may or may not exist in work areas that the operator transfers to.

Each branch, or path, through the decision tree describes a possible scenario that corresponds to a set of sequential operator decisions and specific outcomes at chance nodes. Each given path, if executed, would generate a net reward for the operator over a specific time. Because the system is probabilistic this time and reward cannot be calculated deterministically. However, an expected reward and an expected execution time for each path can be computed. Assuming that a reasonable operator will attempt to maximize his expected reward per unit time then the best decision immediately available to the operator can be chosen.

The results from a decision tree which looks three steps ahead will be optimal if the experiment actually ends three steps into the future. No events further into the future are considered. If, however, the experiment will be continuing for an indeterminate time, then there is no guarantee that some important event will not occur just beyond the three-step horizon. Such an event, if considered in the analysis, might cause an operator to change his "best" strategy.

To include more and more possible events in the analysis, the decision tree can be extended more and more steps into the future. In the limit, as the number of steps becomes infinite, the decision tree will incorporate all possible future developments and the best decision will always be the same as the corresponding optimal Markov model outlined in Chapter 5.

Unfortunately, an infinite decision tree will have an

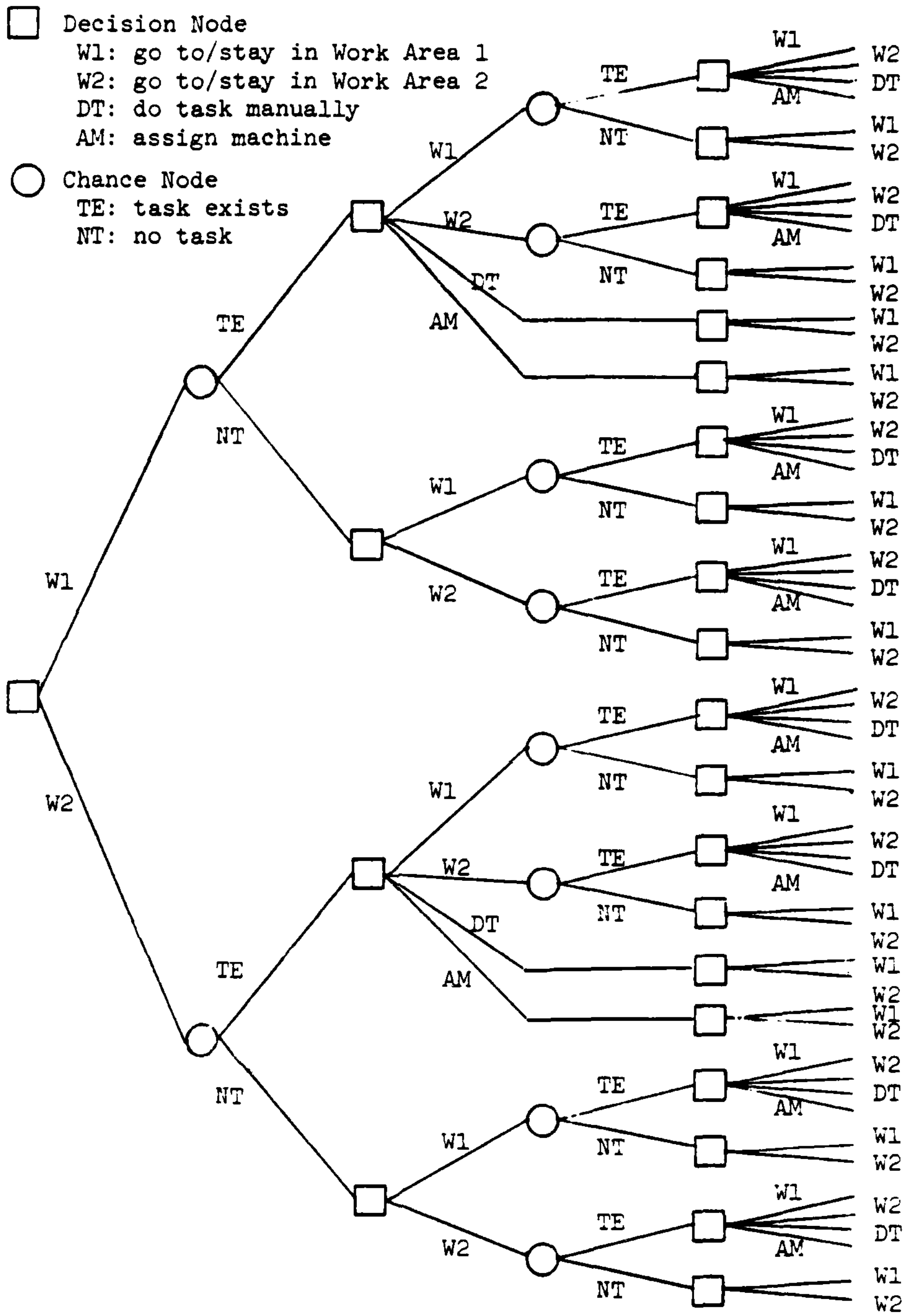

Figure 7.3: The 3-step decision tree for an operator faced with two work areas and having one machine aid. The operator is assumed to start in Work Area 1 where no task exists.

infinite number of branches and the computation of expected reward per time will be impossible. However, as the number of steps gets large, the incremental benefit of looking one more level will decrease. It is possible to find a level of search that is sufficiently small to allow for computation of the "best path", but sufficiently long to approximate the infinite tree.

The procedure described above will not generate the true optimal strategy for the multi-task environments depicted in SUPER because only a limited time horizon is considered. However, it will discover the "best" strategy to apply over that limited time horizon. For the remainder of this chapter, these best strategies will be referred to as "N-step optimal", or "optimal" for brevity.

7.4 Extensions to the N-Step Model

While the N-step model described above can generate the expected reward per unit time of any contemplated action for any level of planning, it may not determine unambiguously the best choice. There will be cases when two actions are equally attractive in terms of reward per time. Some additional citeria must be considered in these cases.

Suppose that in looking only one step ahead, it does not matter whether or not a given task is done manually or automatically: both generate the same expected RPT. A heuristic tie-breaker that could be used in simulations would be to use the machine in such a situation. Such a decision would be rationalized by realizing that a human operator would want to keep himself free in order to explore other work areas, or work on another task while the machine is busy.

Of course, these considerations imply that the decision system is looking beyond the the single step of the model. However, the system cannot be said to be looking a full two steps ahead because it does not specifically say what the operator will do after assigning the machine; rather it deals only with the general possibility that something else might be done.

A hierarchy of operator actions must be prepared for the cases when computed rewards per time are equal. This hierarchy should be a defendable representation of human priorities because the decision process is supposed to model the human operator. For the purposes of this analysis, the operator will first assign his machine aid if available. His second choice will be to do a task manually. Only if these options are unavailable will he change work areas. This hierarchy only applies when two actions cannot be differentiated on the basis of reward per time. The model

will require the simulated operator to change work areas when a task exists only if the reward per time for moving is greater than the reward per time for doing the task.

A difficult problem arises when the model must choose between moving to one of two work areas which have the same RPT over the given planning horizon. A reaonable decision is to transfer to the work area where the probability that a task exists is greatest; that is, all other things being equal, the operator will go to the work area where he feels he will have the greatest chance of actually doing a task.

These heuristic tie-breakers will have a major effect on the model performance when the models are looking only a few steps into the future. The biggest effect will occur in the one-step model. In the one-step model the expected reward of the decision to change work areas does not depend on the operator's chosen destination. Because the model does not look any futher into the future than the time to arrival in the new work area, the expected reward is equal to the sum of the expected holding costs for all work areas over the transition time. This total does not depend on where the operator started his transition or on where the transition will end.

For the one-step model, the decision to change work areas is entirely governed by the heuristic to go to the work are with the highest probability of task existence. The model is essentially reduced to a simple application of heuristics rather than an optimal analysis. This effect, however, is only significant for models which look a very small number of steps into the future. Even for the two-step model, the number of occasions where two decision options generate the same RPT is greatly reduced. Performance is much more depentent on the 2-step algorithm than on the heuristics.

7.5 Pruning the Decision Tree

The number of distinct branches in the decision tree faced by an operator in a multi-task environment depends on how far the operator is looking into the future and on the number of options open to him at each step. If there are R work areas and M machine aids, then there are R+M+1 decisions to be made at each decision node (see Figure 7.2); the operator must decide whether to assign one of the M machines, do the task manually, or go to one of the R work areas. The decision to go to a work area branches into two paths corresponding to the chance that either a task will be found or one won't be. At the final decision node, however, the operator does not consider this chance branching. For a R work area system, and an operator looking N levels deep, the number of distinct branched in the decision tree is

$$\text{Decision tree branches} = (N-1)(2R+M+1) + (R+M+1)$$
$$= N(M+1) + R(2N-1)$$

An example of the proliferation in branches is presented in Table 7.3 for the case of one machine aid in an R work area environment.

	R= 2	4	6	8
1	4	6	8	10
2	10	16	22	28
3	16	26	36	46
N = 4	22	36	50	64
5	28	46	64	82
6	34	56	78	100
7	40	66	92	118

Table 7.3: The number of branches in the decision tree looking at R work areas N steps into the future

The number of branches grows linearly with R, the number of work areas involved. A sophisticated subject might be able to prune the tree by saying that, based on experience, he is extremely unlikely to go to certain work areas in his next planning horizon. He can prune the decision tree by only considering those work areas thet he has a reasonable likelihood of going to.

Such a pruning technique could be applied to the N-step model by having the model look only at work areas that meet certain criteria. The cut-off criterion might include only work areas that have the probability of task occurance above a minimum value, or where the expected reward (probability times reward minus holding cost) exceeded some minimum.

Using a simple-minded criterion to remove possible options is not a sensible way of pruning the decision tree in a supposedly optimal model. The simple criterion might remove options that the optimal model would choose. A more sensible approach is to limit the options actually available to an operator in the environment being modeled.

One approach used in this study was to limit the operator to one machine aid. This reduces the number of branches to 2N+(2N-1)R for the N-step model. Another option involves forcing the operator to follow a given search pattern. The options available to the operator are reduced to doing tasks either manually or by machine, staying in the current work area or moving to the next work area in the search pattern.

By forcing the search pattern, the decision tree reduces to a two work area analysis (the current work area, and the next in the list). The number of branches in the decision tree is therefore $N(M+5)-2$. Limiting the search pattern and increasing the number of machine aids is a sensible approach because this study focuses on the decision to assign machines rather than human ability to generate good search strategies.

CHAPTER 8

MODEL VALIDATION

The N-step decision model is not truly optimal because it has a limited time horizon. The statement was made however, that given a sufficiently large time horizon, the model output would approach the optimal. Before the model can be used as a baseline against which to measure subject performance some effort should be taken to validate the accuracy of this assumption.

8.1 Comparison to the Markov Model

For various situations, optimal performance can be calculated. The Markov model presented in Chapter 5 optimal policies for the case of two work areas and no machine aid. Table 8.1 gives the calculated reward per time for decision trees looking from 1 to 7 steps deep (for a Simulation Confidence Factor of one in a thousand - SCF=.001) as a fraction of the optimal RPT from the Markov model.

N =	1	2	3	4	5	6	7
$\frac{\text{RPT(model)}}{\text{RPT(optimal)}}$ =	.98	.95	.96	1.03	.97	1.02	.99

Table 8.1: Average ratio of RPT for N-step decision models to Markov optimal RPT

All the values of reward per time are close to the optimal value because of the simplicity of the task environment considered. With only two work areas and no

machines even low N decision trees perform well. The reason some of the model RPT's exceed the optimal values is do to the variance inherent in the simulation method.

8.2 Comparing Various Levels of the N-step Search

Given that situations where optimal, or close to optimal policies can be determined, the results of the specific N-step model can be compared to the optimal policies for other values of N. The comparison can take two forms, similar in manner to the way operator performance is computed. The first method of comparison is simply to simulate SUPER using the various trial policies and comparing the total scores. A second method involves comparing specific decisions to the various strategies in various typical situations. The number of aggreements between two models would be a measure of their similarity. Seperate tallies could be maintained to compare models according the the particular type of decision [eg. switching between work areas, assigning machines or deciding to do a task manually].

8.2.1 Comparision by Reward per Time

The validity of a model can be determined through simulation. Over a sufficiently long simulation period the obtained reward per time will approach the long run reward per time of a given strategy. Two policies with similar rewards are similar in effect. Naturally, if a low N model can be found with the same effectiveness as a high N model, it is preferred because of the long computation time required to run simulations with high N models.

A typical series of observed RPT's for models looking up to seven steps deep is presented in Table 8.2. These results are based on 300 second simulation of a four-work-area/two-machine environment.

Depth of Search	Reward/Time
1	3.266
2	3.173
3	3.063
4	3.063
5	3.307
6	3.534
7	3.534

Table 8.2 : Typical reward per unit time as a function of the depth in an N-step search

The general trend shown in Table 8.2 for RPT as a funtion of N is typical of other experimental situations. Reward per time tends to be relatively high for a one and two step search. This is because the heuristics used in the model to decide between two strategies having equal values are effective if not optimal. When three levels of search are considered these heuristics have only a minor impact. Score falls because a 3-step search without heuristics is not very efficient. As N continues to increase though, RPT begins to approach the maximum potential value which would result from an infinite planning horizon.

Occasionally, RPT does not continue to increase with increasing N. In these cases a shorter search strategy just happens to fit the parameters in such a way as to generate very good decisions. When this situation arose, the best policy and not the deepest search was used as the baseline for performance measures.

8.2.2 Comparison by Decision Type

The various levels on N-step models can be compared to each other by looking at the fraction of decisions on which the two models agree. Everytime a subject makes a decision in actual experimentation, the state of the system and the state of the operator's knowledge were recorded. This *information* was used to set up situations in which the N-step models could be compared to one another. Situations from actual experimentation were used because the models would ultimately be used to compute performance for actual experimental subjects.

Table 8.3 shows the number of times a model looking M steps deep agreed with a model looking N steps deep for 253 experimental situations. Table 8.4 looks at the subset of these decisions where a machine was assigned.

		M-step Model						
		1	2	3	4	5	6	7
N-step Mode	1	253						
	2	200	253					
	3	159	205	253				
	4	146	187	233	253			
	5	140	173	215	212	253		
	6	113	142	186	181	215	253	
	7	118	144	181	165	203	223	253

Table 8.3: The number of times a M-step model agreed with a N-step model when all decisions are considered

		M-step Model						
		1	2	3	4	5	6	7
	1	106						
	2	104	149					
	3	103	146	190				
N-step	4	102	144	187	203			
Model	5	82	120	162	170	203		
	6	56	92	132	140	159	172	
	7	55	88	126	130	137	154	173

Table 8.4: The number of times a M-step model agreed with a N-step model when only decisions to assign machine aids are considered

The diagonal of Table 8.4 is the number of times each M-step model assigned a machine. Of the 253 trials considered, the 4-step model assigned a machine 203 times.

For both the above tables, the M-step model best agrees with N-step models where N is close to M. As N differs more and more from M, the fraction of agreements between the two models decreases. When two models have a high fraction of agreement operator performance measured against one will be approximately the same as if the other had been used as a baseline.

The advantages to using the lowest value of N in computing performance lie in the time required to analyse operator strategy. In general, a six-step model was used for most of the analyses of subject performance in this study.

CHAPTER 9

ANALYSIS OF HUMAN BEHAVIOR

9.1 Measures of Human Performance

A human subject makes a sequence of decisions over the course of a single experiment with SUPER. These decisions interact with the particular parameters of the paradigm to yield his score. This score is just a number which reflects the rewards, costs and length of the simulation. It cannot say just how *"good" the operator performed* unless it is compared to some baseline score for the particular scenario. The "N-step" policy model described in Chapter 7 provides that baseline for performance.

The simplest way to measure the operator against the model is to divide the operator's score by the simulation time and compare the result to the reward per unit time (RPT) generated by the optimal model. However, this straightforward comparison could show that the operator's performance exceeds that of the model. The model RPT would be based on a simulation lasting for the equivalent of a few hours real time. Unless the human subject ran for an equal length of time the exact magnitude of his score will depend heavily on the random nature of the task environment. If the human is "lucky" and many tasks arrive over his simulation, then his score will be higher than otherwise. The performance measure chosen should minimize the impact of the variance in RPT resulting from the probabilistic nature of the system.

The problem of variance can be mitigated somewhat by realizing that the random number generator used by the computer in generating chance elements of the SUPER paradigm is not really random. Using the same "seed" on two different occasions will result in the same sequence of random numbers. The same seed used in two separate

experiments will result in the same task arrivals on the same work areas, assuming that a work area is not occupied by a task that has been left unattended when the new task arrival is indicated.

An N-step model can be used to simulate a particular operator over a particular experiment by using the same random number generator seeds. The maximum potential score for both the operator and the model will be identical. However one or the other may miss certain opportunities by leaving a task unattended thereby preventing additional tasks from arriving in that work area.

Even with the same random seed, the score generated by the human subject might still exceed that of the optimal policy. If the cards fall right, the human may always find himself in work areas just as tasks appear. The optimal strategy will, by definition, be the best possible strategy in the long run. It is always possible, though, for the long-run optimal strategy to be significantly sub-optimal in a variety of short term situations.

A third possible measurement of performance involves looking at individual decisions made by the experimental subject. The game conditions at the time of these decisions are made can be recorded. The best-choice decision based on an optimal N-step strategy can be computed for these conditions. The fraction of human decisions that agree with the computed best strategies is the third measure of operator performance.

This performance measure is based on what an optimal operator would do when faced by the exact decision environment actually faced by human subjects. The measure differs from the previous two in that both the actual performance and the baseline it is compared to are computed for identical conditions.

An additional advantage of this performance measure is that it exposes which types of decisions are leading to human sup-optimality. The fraction of operator decisions that agree with the optimal model can be computed by decision type. The number of times a subject used a given machine when the optimal model did not, and the number of times the human did not when the model did, are readily computable quantities that provide specific information not available in gross performance measures.

Breaking down performance by decision type also allows this study to focus specifically on the way human subjects assign machines, as opposed to the effectiveness of the search strategies they employ in moving from work area to work area.

The N-step model is actually a series of models each of which looks one additional decision step into the future. A comparison of human to model performance performance can be generated for different levels of N in the N-step model. There will be certain levels of N that yield the highest correlation with human performance. This exercise gives an indication of the level of forecasting used by the human operator.

9.2 Some Reasons for Operator Sub-optimality

If subject performance does not match the model output, then the human operator can be called sub-optimal. However there are many possible explanations for this sub-optimality, only some of which support the hypothesis that the human is being inefficient in the allocation of his resources.

9.2.1 Non-Linear Utility

One such possibility is that the subject's utility for points he earns in an experiment is not linear. A subject's responses might be optimal in terms of his internal utilities while appearing sub-optimal when compared with results based on a model which assumes linearity. The hypothesis that subjects have non-linear utility maps was the subject of Chapter 6. The conclusion was that linear utility holds reasonably well.

Utility deals with the internal value subjects place on the objectives in the experimental paridigm. There are other areas in the paradigm where the difference between the actual game and the subject's internal representation of it might lead to sub-optimal behavior.

Consider the situation in SUPER. The human operator must search the work space for tasks requiring attention. Both the search strategy and the method chosen for dealing with tasks depend on the various parameters involved.

9.2.2 The Operator's Internal Representation of Task Parameters

Technically, the operator's decisions do not depend directly on the task parameters. Instead those decisions depend on the subjects internal representation of those parameters. Suppose the subject knows nothing about the tasks he is to face. His best course of action will be to assume that all tasks are identical. The operator will act optimally according to the information he has available, though his behavior will appear sub-optimal to an observer who knows the correct task parameters.

In the above example the subject was assumed, for whatever reason, to have no knowledge of the system in which he must operate. In the experimental situation, the subject was presented with all the relevent information about the paradigm beforehand. He also developed a sense of the parameters as he was exposed to them over the course of the experiment. His actions depended on both the decision strategy used and his perceptions of the parameters.

In general, a subject will accurately encode only some fraction of the information presented him. Some information he will forget, and some information he will make up to take the place of forgotten information. The information link between the real world and the subject's decision processor is not transparent.

Figure 9.1 graphically presents this process. The subject is presented with the task parameters and encodes then into his internal representation. The fact that this process is not perfect is represented by the misalignment of the box signifying his internal representation (IR) and the box for task parameters (TP).

The encoding of task parameters is an ongoing process. When a person is involved in such a multi-task environment, his opinions vary with his experience. If one particular task class seems to be cropping up often, then the operator might adjust his concept of the arrival rate for that task. Similarly, if the subject is getting no feedback concerning his performance, his internal representation of the distinction between the various rewards and costs of differing tasks will be eroded. The agreement between the actual task parameters and the subject's internal representation is not static; it will shift and wobble depending on his memory and his experience. At any point in time, however, he will have some concept of all the parameters that are influencing his decisions.

Based on his internal representation of the working environment, an operator will determine a course of action which may not agree with the optimal model. Figure 9.1 presents two hypotheses for this sub-optimally. First, the mental computations performed by the operator may be such that he does not choose the best actions. Second, if the subject's computation algorithm is optimal, sub-optimal performance can result from an internal representation that does not agree with the actual task parameters. These two hypotheses are not exclusive. The subject may be imperfectly encoding information and then processing that information sub-optimally.

The optimal model bases its calculations on the actual task parameters. The human operator bases his computations

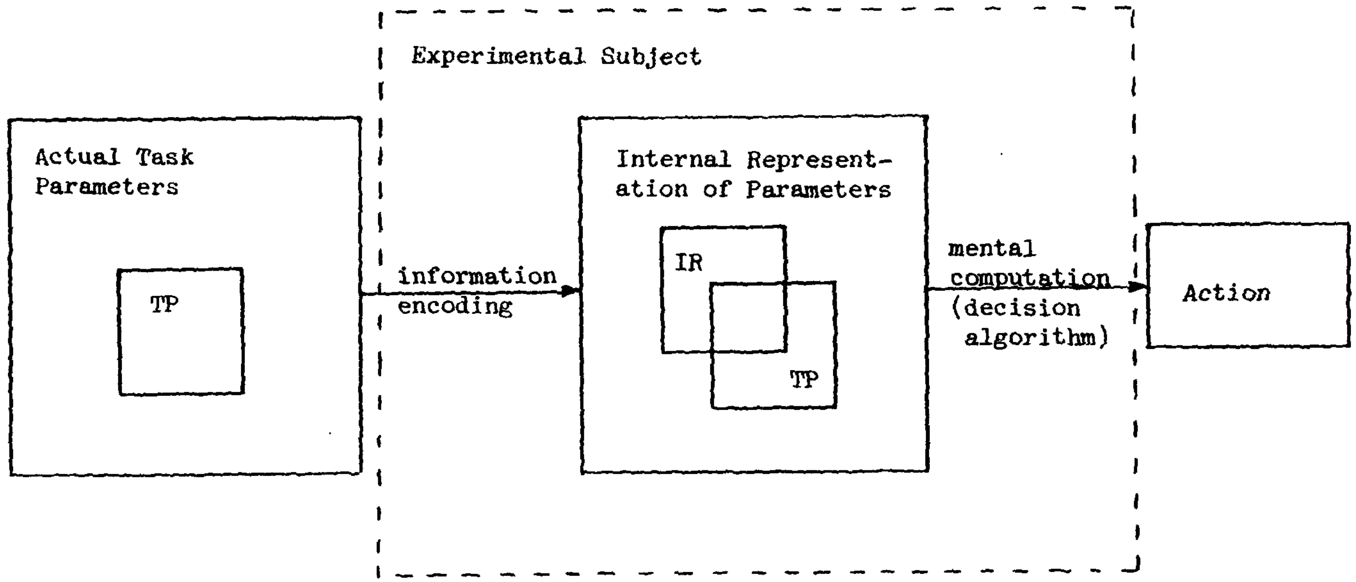

Figure 9.1: Representation of a subject encoding task information for use in his decision algorithm ultimately leading to action

on his internal representation of those parameters. To the extent that the internal representation does not agree with the task parameters, model and human behavior will differ.

If the operator is indeed making satisfactory decisions based on his concept of the world, then the system designer must focus on improving the operator's internal representation in order to achieve desired performance. However, if it is the operator's thought process that is leading to inefficiencies, there may be nothing that can be done to improve performance.

This report is concerned with deficiencies in the decision algorithm used by human operators rather than with problems they have encoding information. To reduce the discrepency between the internal representation and the actual parameters for experimental purposes, two strategies were used.

The first strategy involved directly aiding the subject in encoding information by placing the task parameters in the context of a story, or scenario. One such scenario involved thinking of the various task classes in terms of using a calculator. One task class was labelled "Additions". Tasks in this class had low rewards, high arrival rates, and took about the same time for the operator to do as it did his machine aid (i.e. his calculator). A second class was labelled "Exponentiation". Exponentiation tasks were much rarer than Additions (i.e. they had a lower arrival rate) and there was a significant advantage to assigning the calculator/machine-aid to them instead of dealing with them by hand.

It was felt that by presenting each task class in terms of a coherent story the operator would be able to compare the parameters between tasks by simply thinking of their names. Results with this strategy were promising in that subjects were able to recall task parameters better than when no story was presented. However, this improvement may have been due to the fact that subjects spent more time studying the task parameters when the story was presented than they did otherwise. That is, the exercise itself of learning the story provided more of a benefit than the use of the names as memory aids.

A second method of reducing the difference between internal representation and actual task parameters was to display a summary sheet listing all the task parameters on the Megatek display beside the playing field. If the subject needed any information he could consult this summary instead of relying on his memory. Operators also found the summary sheet useful in comparing two or more distinct task classes.

9.2.3 The Mental Computation Stage

The subject's decisions may be said to result from some mental computations performed on his internal representation of task parameters. There is a pre-processor in this mental computation stage that is not specifically part of the decision algorithm. This pre-processor places subjective values on the various parameters found in his internal representation.

Utility for costs and rewards, which has been dicsussed at length elsewhere, is one example of this preprocessing. The subject's internal model may be an accurate representation of the task parameters but he may values these parameters differently than does the optimal model.

The results in Chapter 6 indicate that the utility for points in SUPER in appear linear, as was assumed in the N-step model. However, there are similar judgements concerning how the subject perceives the probability that tasks exists on other work areas. All the subject has is an idea of how much time has elapsed since he last visited those work areas, which he somehow converts into the probabilities used in his decision algorithm.

There is no way to separate the effects of the pre-processing of the internal representation from sub-optimalities resulting from the subject's decision process. Whatever the source, these sub-optimalities are the direct result of the human mental process and are important to this study.

$$\begin{tikzcd} \text{Task Parameters} \arrow[d] \\ \text{Internal Representation} \arrow[d, "\text{pre-processor}"] \\ \text{Utility for Parameters} \arrow[d, "\text{decision algorithm}"] \\ \text{Action} \end{tikzcd}$$

Figure 9.2 :- Movement from task parameters to operator action

9.2.4 Operator Reaction Time

An additional reason that an operator might not agree with an optimal model even if his mental computations followed the same optimal algorithm is that the human requires a finite reaction time to respond to stimuli. The N-step model described above assumes that upon arrival in a work area, the operator instantaneously perceives the situation and makes a new decision. The elapsed time spent in work area can be zero in the model. A human does not respond in this way.

The average time between a subject's arrival in a work area and the time when he takes some action, such as assigning a machine or leaving the work area, is .48 seconds (based on 3000 decision trials for four different subjects). This value will vary by subject because some operators are naturally quicker or are more willing to make a hasty decision that they might later decide was inappropriate. A particular subject will vary his reaction time between experiments depending on the number of decision options open to him and on the complexity of his mental algorithm.

It is possible to measure a subject's reaction time and incorporate it into the N-step model. A self-aware subject will realize that he requires .5 seconds to make decisions. This reaction time can be added to the expected time computations for the N-step decision tree. When a subject's performance was calculated, the baseline model included the subject's observed reaction time. If this were not done then the subject and the optimal model would be dealing with slightly different situations and sub-optimality could be attributed to this difference.

9.3 The Experimental Situation

This study focused on sub-optimalities in operator performance originating in the operator's mental computations rather than in the other areas mentioned above. The script for each experiment run reflects this concern.

9.3.1 The Task Familiarization Phase

Each experiment was preceded by a Task Familiarization Phase. Here the subject was exposed to each of the task classes that he would face in the subsequent experiment. Each work area was displayed individually on the Megatek screen (see Figure 9.3). Tasks assosciated with that work area were allowed to appear at their given arrival rate. The operator's options included assigning machines or doing task manually; he was not allowed to change work areas. A score based solely on the operator's performance in the single work area was displayed.

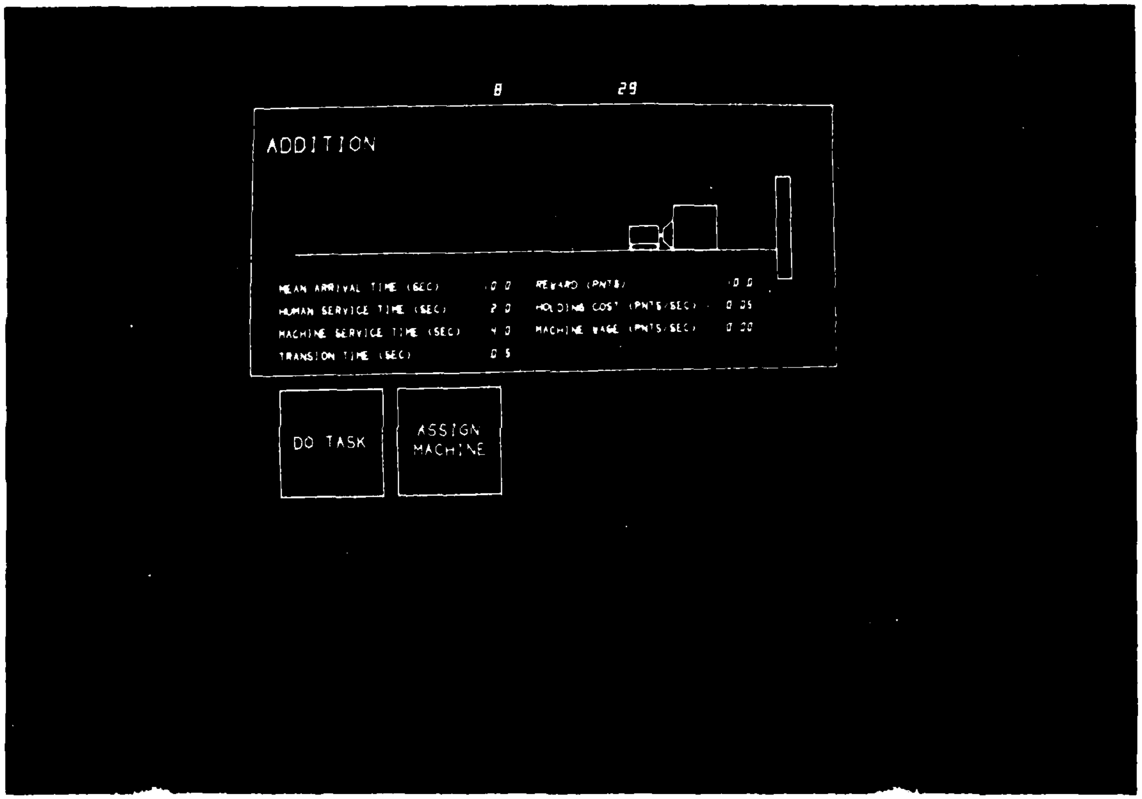

Figure 9.3: Megatek display during the Task Familiarization Phase of experimentation

By forcing the operator to stay in one work area, it was hoped that he would familiarize himself with the parameters associated with that work area. In the actual experiment a subject could jump from work area to work area and cannot readily acquire a feel for the interarrival times and machine service times involved. Because the displayed score in the Task Familiarization Phase was for training purposes and was not recorded anywhere, the operator could, in an unpressured environment, see how various methods of dealing with that task class affected his score.

A summary of the task class parameters was displayed on the Megatek thoughout this stage for reference purposes. When the operator felt he had absorbed the information for one work area, he could then look at another one. At any point the operator could return to any work area for a "refresher course" in those task parameters.

The purpose of the Task Familiarization Phase was to help the operator accurately encode, and develop a feel for, the task parameters. If there was a "story" that went along with a particular experiment, it was explained at this point. When the operator felt he has encoded all the information he could start actually playing SUPER.

9.3.2 Warm-Up Phase

In the Task Familiarization Phase, the operator only looked at one work area at a time. The only options he could test were the use of different machines on specific tasks. He could not examine his entire strategy which included switching between many work areas.

As a precursor to the actual experiment, each subject was allowed a warm-up period of about two minutes in which his score was displayed to provide him with feedback. The Warm-Up phase also served to remove subject start-up (learning curve) effects.

Additionally, when a run with SUPER was started all work areas were empty and the operator knew this. There was never be another situation in the same experiment when the operator knew all work areas were clear. This start-up condition was an anomoly and should not be counted in experimental results. The Warm-Up Phase allowed SUPER time to reach typical conditions.

9.3.3 The Experimental Phase

The Experimental Phase began when the Warm-Up Phase ended, without reinitializing SUPER. The operator was signalled that the experiment had started and his actions were being recorded both because of the time (he knew the

length of the Warm-up Phase) and because his score was no longer displayed.

The score was suppressed because it could provide the operator with information that would normally be unavailable. If the operator was in an empty work area and suddenly saw the score begin to fall then he could assume that a task had appeared on another work area and was then exacting a holding cost. Similarly, the rate at which score decreases indicated the number of tasks available. This information could be of use the the operator, but is unrealistic to provide him with it.

Everything which occured in SUPER over the course of an experiment was recorded in a data file. All task arrivals, task departures and operator actions, and the times these events occur were stored. This information could be accessed later in order to reconstruct the state of the system and the potential state of the operator's knowledge at each decision point. This reconstruction of the operator's decision environmnet was used in the N-step optimal models to calculate performance.

9.3.4 The Debriefing Phase

After each experiment the subject was debriefed. The debriefing, while generally unstructured, progressed according to the following outline.

First the operator was asked to recall all the task parameters from the recently completed experiment. No prodding was given for this initial pass. Values that the subject was uncertain about were then explored by asking the subject what he would have done in situations where those parameters should have been important. This questioning helped subjects recall those parameters.

The second stage of the debriefing asked the subject to define the strategy he used in the experiment. Questions were asked until the strategy was well-defined enough for the experimenter to be able to determine what the operator would do in any situation.

The operator was also asked to comment on how well he felt he had followed his stated strategy. Experiments where the subject found a good strategy, but did not begin to apply it until the end of the run, were repeated.

Finally, the subject was asked to resolve differences between his stated strategy and the values of the parameters he recalled. Many times the operator could not recall any difference between two work areas in the initial phase of the debriefing but would have a strategy that clearly

preferred one over the other. Often by considering his strategy, the operator could improve the accuracy of his parameter recall.

The purpose of the debriefing stage was to generate a picture of the subject's internal representation of the task parameters. The recalled parameters may not necessarily agree with the subject's actual internal representation but they do give an indication of weaknesses in the internal representation.

Suppose there are many errors in a subject's recall. This would indicate that the subject's internal representation of those task parameters is also in error. Uncertainty about the difference in a particular parameter between task classes implies that that parameter did not factor into the subject's decision algorithm. Similarly, parameters that the subject accessed often would be better remembered. Not surprizingly, inexperienced subjects, who used the simplest decision algorithms, were also the ones who exhibited the greatest uncertainty in recalling task parameters.

The Debriefing Phase provided feedback to the experiment designers who were trying to generate an experimental situation that facilitated the subject's ability to encode task parameters. Through the Task Familiarization Phase and by displaying task parameters, subject recall could be close to perfect. This implies that any subject sub-optimality could be attributed to the subject's mental decision algorithm rather than to his ability to encode task parameters.

CHAPTER 10

EXPERIMENTAL DESIGN

10.1 Choosing Task Parameters

The various phases in the experimental situation, from task familiarization to subject debriefing, were described in Chapter 9. What has yet to be discussed is the choice of task parameters and machine productivities that were used in the experiments.

Prior to the period when specific hypotheses concerning man-machine interaction were tested using SUPER an extensive series of training trails were conducted. These trials served the double purpose of training subjects in order to minimize learning curve effects in actual experiments, and exposing the experimenters to any potential difficulties that they might have to deal with in running those experiments.

A key conclusion from this training period was that subject performance depended on the speed at which events occured. Each experiment has a characteristic time interval, which will depend on the average interarrival times of task and their service times. If this characteristic time interval is scaled proportionally the strategy determined by an optimal model will not change.

An example of such a proportional scaling would be the doubling of all parameters that vary with time. These parameters are the arrival rates, service rates and holding costs, all of which are defined in terms of events or costs per unit time. This doubling of the parameters exactly doubles the long run reward per unit time generated by the N-step model.

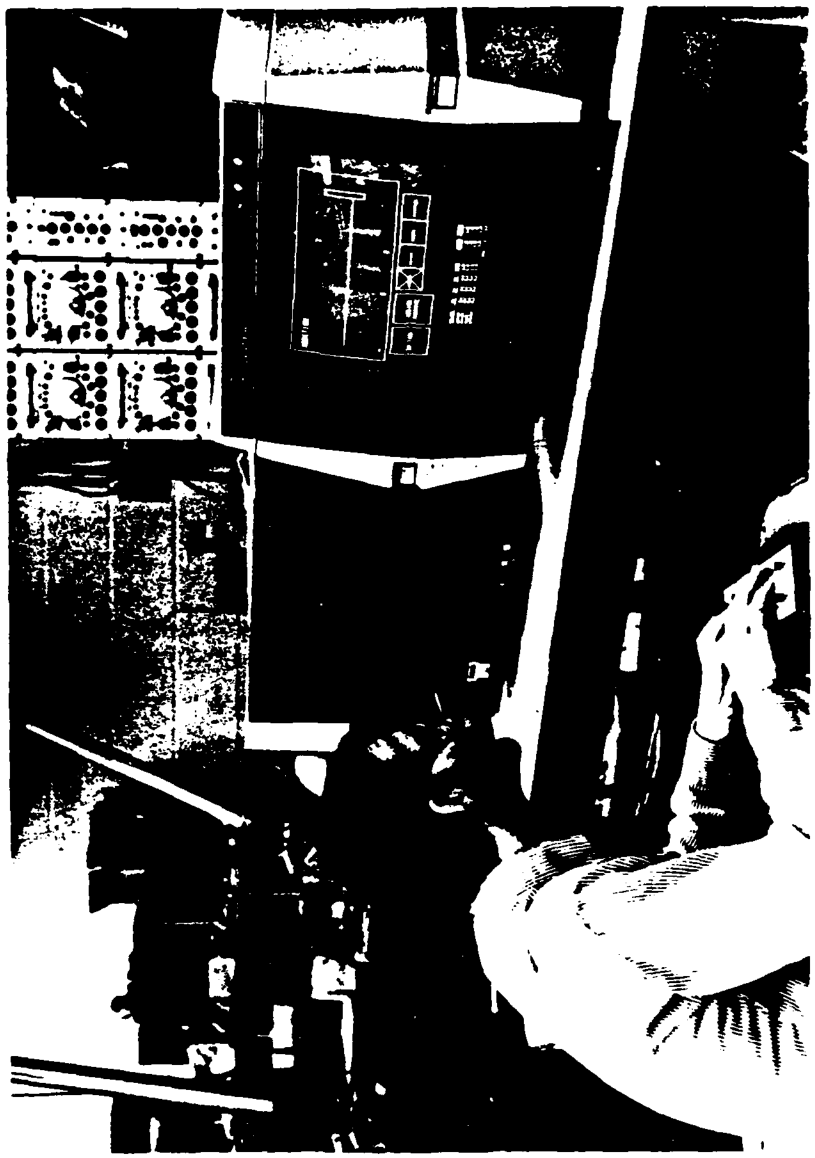

Figure 10.1: Photograph of subject absorbed in an experiment with SUPER

When human subjects were faced with a similar doubling of parameters, they were unable to double their reward per unit time. Human beings can be modeled to a first approximation as single channel information processors. That is, they can only be doing one higher order mental process at any instant. Subject were required to perform two distinct tasks in SUPER.

First, subjects had to gleam information from the Megatek display. Everytime they change work areas they had to see if a task existed there, and if so whether a machine was assigned there. They should also watch for task arrivals and task completions on the displayed work area. Second, the subjects had to make decisions about when to change work areas and how to deal with discovered tasks. Subjects could be assumed to be allocate their mental processor between the competing demands of display scanning and decision making.

When the characteristic time interval in SUPER is shortened the rate at which significant events occur increases. More tasks arrive and the time required to complete those tasks decreases. Subjects need to spend more time scanning the display to keep up with the changing events. Less time is available for decision making.

When the time available for decision making begins to impinge on the time required for subjects to implement their decision algorithm performance suffers. Subject will be forced to use a simpler algortihm, one that varies less with changing circumstances.

Increasing the characteristic time interval should produce the complementary effect of improving performance. The rate of improvement will eventually fall to zero as the time available for decision making excedes the time required for subjects' most complicated algorithm. After this point, any additional time will not imporve the subjects' strategy. It was hoped that all the experiments run in this study would have characteristic time intervals that allow the operaters to use their best algorithms.

10.1.1 Determining Satisfactory Characteristic Time Interval

A simple way of testing whether a particular experiment had a sufficiently large characteristic time was to double it and then repeat the experiment. If a subject's performance did not improve in this second round then it could be assumed that the original time interval did not force the subject to make any quick decisions. Unfortunately, applying this test everywhere would have tripled the time required to run experiments since each

test, with the longer time interval, would have taken twice as long as the original experiment.

An additional result from the trial phase of SUPER was that subjects could tell, subjectively, whether or not they were "comfortable" with a given set of task parameters. If subjects always felt comfortable when the time interval was long and rushed when it was too short, the comfort criterion could be used to determine satisfactory parameter choices.

Three scenarios, involving one machine aid and two-, four- and eight-work areas, with unamimous subject aggreement on comfort were found. These versions were defined as "normal". Additional versions with characteristic times double the normal and half of normal were also created. These parameter sets were termed "slow" and "fast" respectively.

Three subjects were each brought into the laboratory for three separate sessions. In each session three experiments were conducted covering each of the three scenarios and each of the three speeds. Order was varied from subject to subject to soften learning curve effects. Table 10.1 lists the average actual performance, measured as the fraction of decisions that agreed with the optimal model, for all three subjects over the different versions of each scenario.

	Version		
Scenario	Fast	Normal	Slow
2 work areas	.75	.82	.72
4 work areas	.60	.62	.48
8 work areas	.76	.91	.56
Average	.70	.78	.59

Table 10.1 : Average performance relative to an optimal model for three subjects

Compared to the "normal" scenarios. performance was significantly worse for the "fast" scenarios where the characteristic time was shortened. It is interesting that "slow" performance was actually worse than "normal". This decline is most attributed to the excess time subjects had after they had scanned the screen and made decisions. Because they had nothing to do they got bored and disinterested. This boredom was comfirmed by subject comments in the debriefing phase of experiments. Also, because of the long periods of inactivity subjects forgot what work areas they had resently visited and their search

algorithm suffered.

The absolute difference from scenario to scenario is due to the different task environments. In the eight-work area scenarios all tasks were identical. Search strategy therefore only vary with elapsed time since a given work area was last visited and not according to rewards and costs. Because it is easier to deal with, performance was better than in the two- and four-work area scenarios where task classes differed and the operators were faced with a more complex environment.

Performance is better in the two-work area scenario than in the four because two-work areas are less complicated than four. In the former the destination for a transition is established as the other work area. The operator is only concerned with when to switch, not where to switch to.

10.2 Finding Instances of Human Sub-optimality

The focus of this study is on areas where human decisions to assign machine aids are sub-optimal. The trial period with SUPER was also used to scout for such areas that could be explored in depth in the actual experimental phase.

In the trial phase, task parameters and the number of machines were varied substantially from run to run. Subject performance, particularly in the area of assigning machines was examined and the results used to develop hypotheses in need of further experimentation.

CHAPTER 11

EXPERIMENTAL RESULTS

11.1 Introduction

In man-machine systems considered in this study there were two distinct aspects to the operator's supervisory role. The operator first had to search for tasks. Once a task was found a method of dealing with it had to be determined. An operator's performance in determining a search strategy was not necessarily dependent on his performance in deciding whether to do tasks manually or to assign machine aids.

This study was chiefly concerned with the decision to assign machines. However, the nature of the experimental paradigm provided a mechanism for exploring the search strategies of operators and certain conclusions can be drawn in this regard.

11.2 The Operator's Search Strategies

Depending on the values of task parameters such as mean arrival rate and service times, approximately seventy percent of all the decisions made by subjects involve changing work areas. Determining where to go next was a problem faced by the operator very frequently. However, subjects did not always make the best decisions.

11.2.1 The Effect of Transition Time

The importance of the decision to change work areas was dependent on the transition time between work areas. If the transition time was very short a mistake could easily be corrected. In the limit, a zero transition time is similar to being able to examine all work areas simultaneously and having perfect information about task occurances.

As transition time increases making the right choice about where to go becomes more and more important. Ultimately, however, it will be best not to move at all because transitions take so much time. Whether a given value for transition time is "long" or "short" depends on the characteristic time interval (a measure of mean arrival rates and service times) for the system.

The effort SUPER subjects put into the decision to change work areas did vary with transition time. For short transition times subjects employed only the simplest search algorithms. In the debriefing phase comments such as "I had no real plan; I just went somewhere else" or "I just cycled through the work areas in order" were commòn. These curt remarks indicated subjects placed only minimal importance on search strategy.

When subjects did care about their search algorithm they could be very lucid in decribing it. Debriefings were full of conditional statements along the pattern of: " If I were in work area B and just assigned a machine I would go to A unless I had recently completed a task there in which case I would go to C." The time spent elucidating on search strategy varied directly with transition time.

Subject performance, as measured by fraction of operator decisions that agreed with the N-step model, also varied with transiton time. The optimal model always put the same effort into its choices regardless of the value of transition time. Subjects did make more mistakes when they cared less about the consequences. However, the occasions on which they cared less about the consequences were those occasions when the consequences mattered less. Short transition time mean that mistakes could be corrected quickly and the impact of an incorrect decision could be minimized.

In most real life situations the transition time between work areas is relatively short when compared to task interarrival times and service times. A pilot scanning displays looking for problems is limited by how fast he can shift his glance or move his hands to necessary controls. A plant supervisor must only take the time to walk from machine to machine. Exploring subject behavior for long transition times was not undertaken because, while interesting, behavior would not have any relevance to reality.

11.2.2 Human Information Processing Limitations

The search strategy employed by human subjects depended less on the current condition of the experimental system than did the model strategy. This is only saying that

subjects tended to develop a search pattern which they employed consistently instead of analysing the decision to change work areas everytime they encounted it.

The N-step model has no memory. Every time it is faced with a situation requiring a decision it will grind through its optimizing equations regardless of how many times in the past it has faced identical, or near-identical, situations.

A human operator can recall similar situations he has dealt with previously. Additionally he does not have the information processing capabilities of a computer. It is not surprizing, therefore, that human subjects develop a simple search algorithm which they believe has worked well previously.

In the previous section instances were discussed where subjects used complicated conditionals in determining where to transfer to. While such algorithms did depend on the state of the system they did not explicitly deal with all possible parameter values. They seldom took more into account than the last states visited and where the last few tasks were found.

The search strategy chosen by the operator could be altered by the experimenter simply by changing the buttons on the control box which corresponded to the differing work areas. For cases where there wasn't a big difference between tasks and where transition time wasn't too important the subject's search strategy would often be to press buttons in sequence until a task was found.

The fact that geographic proximity of two work areas affected search pattern will be of no surprize to any control panel designer who worries about the effects of display position on performance. However, the fact that search strategy could be so easily manipulated lends credence to the hypothesis that human operators will sacrifice performance in favor of an easily implemented strategy.

11.2.3 The Effect of Subjective Probability Estimation

Deciding where to go next will depend on the operator's belief that a task exists at his intended destination. There is substantial evidence in the human factors and psychology literature (e.g. Tversky [26,27]) that implies that humans overestimate the liklihood of low probability events and underestimate it for high probability events.

Subjects given experimental situations with one task class having a very low arrival rate visited that work area much more often than specified by the optimal model. Even

when rewards and costs were the same for all work areas subjects visited a work area with an interarrival time of 100 seconds every twenty-five seconds and a work area with a ten second interarrival time every five seconds. The optimal model did not visit the former except at fifty second intervals.

A very low arrival rate means that a work area should not be visited too often. Experimentat subjects tended to develop search strategies that boiled down to a straight pattern for the more active work areas with transitions to the low arrival area treated as an afterthought.

Other factors beside probability of task occurance, such as reward and service time, affect the relative desirability of work areas. If any general comment can be made about the way human subjects decide where to go, it is that they tend to visit the less desirable areas more than optimal.

11.3 Choosing How to do Tasks

Most of an decisions faced by operators dealt with the question of where to go next. However, the major component of performance was how they chose to deal with tasks they did find. The decision to do a task manually or to assign it to a machine was the central question dealt with in this study.

11.3.1 Machines That are More Productive Than Their Supervisors

If a machine can complete a given task class much faster than the human operator there is no question that the machine should always be assigned to that task. Not surprizingly, human operators were willing to acknowledge a machine's superiority and were willing to take advantage of it.

There are a variety of ways to model a productive machine that have differing effects on human behavior. A machine can be very productive at a given task because it is specialized. In this case the machine is also very unproductive at other task classes. Subjects faced with such a situation assigned machines to those tasks where the machines were effective and would do the other tasks manually.

In such a situation human performance in assigning machines was close to optimal. Any overall sub-optimality was due to poor search algorithms. In general, though, search strategies were very good with subjects first checking those work areas where a machine was clearly

preferred before undertaking any tasks manually. In this way they could maximize the time when both the machine and the human were working concurently.

A second way of modelling a productive machine was to assume that it was better than the operator at all tasks. Subjects would then move from work area to work area assigning the machine whenever a task was found. Because the machine was so productive it was usually free by the time a new task was found. If not the subject would wait for it to become available before proceeding.

Subjects behaved close to optimally when they had such a superior machine because the machine was so obviously effective. The operator had to struggle to find tasks to keep the machine busy. As long as the machine was used whenever a task is found opportunities to behave sub-optimally were limited. This scenario was also slightly unrealistic because such a superior machine should be able to find its own problems in reality.

An interesting, though slightly unrealistic case, was where the machine was very good at certain tasks but only average at others. The operator needed to decide whether or not to assign the machine to one of the average tasks knowing that this might tie up the machine and prevent it from being used where it is clearly superior. However, it is hard to come up with an example from real life which corresponds to this choice of parameters.

Subjects reacted to this scenario by never going to the work area where the machine was superior unless the machine was free. Otherwise they would assign the machine to other tasks and hope to find another average task to do concurrently, though manually. This strategy yielded very good results in terms of performance based on decisions to assign tasks. The search pattern chosen by the operator determined just how close to optimal he came.

In summary, for the case of highly productive machines human operators did make an effecient allocation of tasks between themselves and their machine aids. Any deviation from optimal can be traced to the search strategy employed or to errors subjects made in applying their strategy.

11.3.2 Men and Machines with Comparable Abilities

Many situations exist in reality where machine aids perform tasks in roughly the same manner as the men they replace. The flight computer in an aircraft acts like a co-pilot who doesn't get bored; it does not fly the plane any faster or more efficiently than the human pilots. When men and machine aids have the same productivities for tasks

it is reasonable to assume that they both carry out the same steps in dealing with tasks. As a result the man and machine are directly interchangable and an operator may start a task himself and then assign it half completed to his machine aid.

When the experimental situation was set up to reflect interchangable men and machine aids with similar productivities human subjects performed very well. A standard strategy was to search for a task and then assign it to a machine aid. If no machines were available then the operator started on the task himself until a machine became free. The N-step model generated an almost identical strategy.

A few experiments were conducted where theman and the machine aids were not interchangealbe. Tasks started manually could not be transferred to machines. In these cases the subject would again assign the machine if available and then do secondary tasks manually. If the subject felt that other tasks existed on other work areas he might stop doing the present task, move to the other work area, assign the machine there, and then return to the unfinished task. Subjects did this less than would be indicated by an optimal model but because occasions for this behavior seldom arose performance was not affected much.

11.3.3 Assigning Low Productivity Machines

The use of low productivity machines is often an option in industrial systems. Utilities maintain expensive oil burners to deal with periods of peak demand. Similarly manufacturers retain old machinery to deal with contingencies such as new machinery breakdowns or unexpected orders. It is not unreasonable for a system designer to supply a human supervisor with an inefficient but cheaply designed machine to provide emergency system capacity.

The following set of experiments is illustrative of results obtained concerning human operator use of low productivity machines. In these experiments an operator was given two machine aids. One of these machine aids had the same productivity as the operator in all work areas. The productivity of the second was scaled down by a factor of X. When X was one, the second machine was identical to the first.

Figure 11.1 shows average subject performance as a function of the relative productivity of the less productive machine. Two performance measures are used. The first is the fraction of times, when faced with a task in the current work area, that the operator chose the same strategy as the best N-step model. The second measure is a subset of the

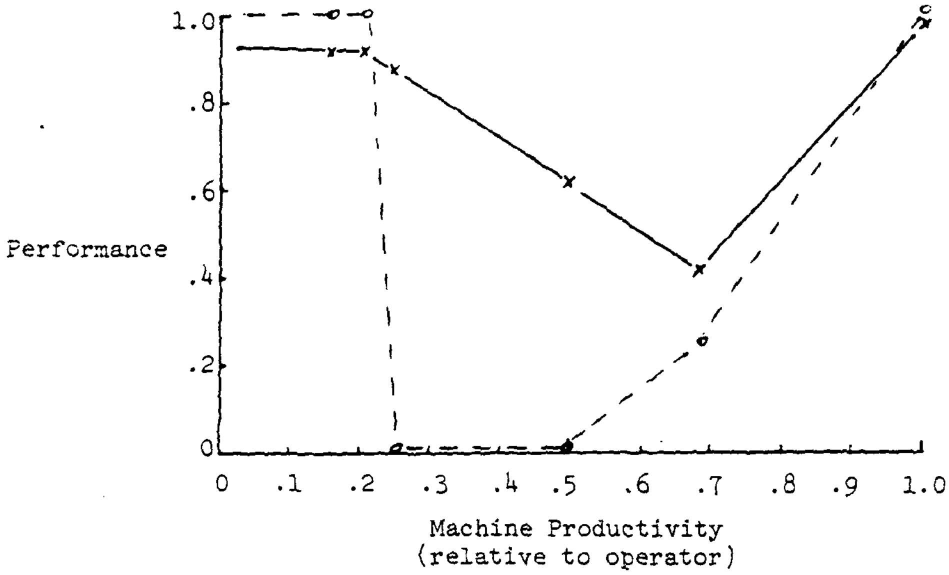

Figure 11.1: Operator performance as measured by the fraction of times the optimal model agrees with the subject's decisions for
—x— all decisions
--o-- only those decisions assigning the low productivity machine

AD-A118 419 MASSACHUSETTS INST OF TECH CAMBRIDGE MAN-MACHINE SYS--ETC F/G 5/8
THE USE OF MACHINE AIDS IN DYNAMIC MULTI-TASK ENVIRONMENTS: A C--ETC(U)
JUN 82 W T WOOD N00014-77-C-0256

UNCLASSIFIED NL

2 of 2

END
DATE FILMED
09-82
DTIC

first, and is the fraction of times the model agreed with the subject when he assigned the less productive machine.

When X, the relative machine priority, was one human performance is close to perfect by both performance measures. However, decreasing X reduces performance drastically. The second measure shows that this reduction is due to the fact that the model never (for X less than .5) uses the low productivity machine. The human, however, resolutely sticks to his old friend and doesn't abandon this machine until its productivity has dropped down to one fifth of the human. Ususally when the human wanted to use the low productivity machine, the model specified that the task be done manually.

The point was made to subjects that they did not have to use the machine if they didn't want to. Yet Figure 11.1 shows that they did. When questioned later, subjects seemed to feel that they would be wasting some of their resources if they did not use the machine.

A common explanation by subjects for behavior was that they thought they should assign a low productivity machine to some task they wouldn't get to often. They could thereby reduce the number of work areas they had to deal with by one, at least until the poor machine was finished. While the poor machine was working the operator would only be working with his good machine on a reduced system. Making the system simpler by assigning a slow machine on a work area and then forgetting about it seemed to be a sensible goal to many subjects.

Another explanation is that by using the low productivity machine it was possible to be working on three tasks at once: two by machine and one manually. Subject behavior in many diverse situations could be explained by assuming that subjects wanted to have as much going on at once as possible.

Subject performance, from Figure 11.1, improves again when subjects finally decide to give up the slow machine. At this point subjects are moving from work area to work area doing tasks manually only when the fast machine is busy.

The effect of slow machine productivity on strategy is shown in Figure 11.2. The fraction of times that subjects and the optimal model would assign the low productivity machine to do a task is plotted against that productivity for the same set of experiments used to generate Figure 11.1.

For X equals one both machines are equally effective.

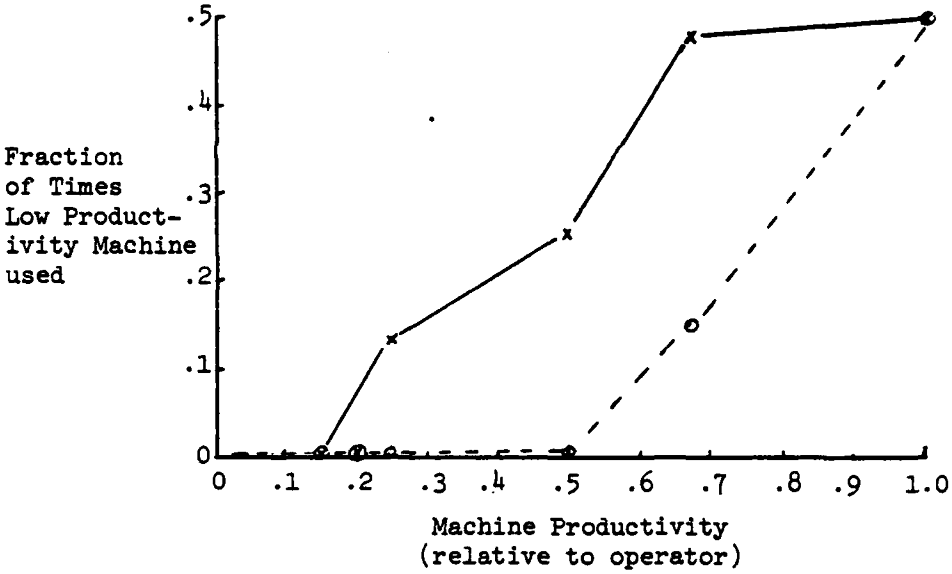

Figure 11.2: Fraction of all decisions to deal with a task where the low productivity machine was used - for decisions made by
—x— experimental subjects
- - o - - the optimal model

With two such productive machines the operator did not have to deal with any tasks manually. The likelihood of using any one machine for X equal one in therfore 0.5 as is shown on the graph. The optimal model gives up on the slow machine almost immediately while the subject continues to use it.

11.3.4 The Impact of Machine Wages on Strategy

Decreasing a machines productivity will make it less desirable to use. Increasing its wage whould have the same effect. The net reward for completing a task by machine equals the final reward minus the service time times the sum of holding cost and machine wage.

$$\text{Net Reward} = R - (h+w)/u$$

where u is the service rate.

When the holding cost is zero, a doubling of the service time (1/u) will have the same effect on net reward as doubling the machine wage. Both should make the operator more hesitant to use that particular machine on that particular task. If either should have a greater effect it should be the service time. Using a slow machine will tie up a work area for a long period of time. No new tasks will appear there and the operator will be unable to realize their potential rewards. Based on the observed impact of increased service times on strategy, a reasonable hypothesis would be that increasing wages would have only a minor effect on subjects' decision to use machines.

This hypothesis was tested for the same task environment as was used to generate Figures 11.1 and 11.2. There was no holding cost and the operator was given two machines with identical productivities. The wage of one machine was fixed at one point per second while that of the second varied up to five points per second.

Figure 11.3 shows subject performance as a ratio of the two wages. Both overall performance in dealing with tasks and the fraction of times the model agreed with the operator when he assigned the expensive machine are plotted.

Performance is much better than observed in the case of decreasing productivity. Surprizingly, high wages had a large inhibiting effect on the subjects' use of the expensive machine. In fact, for a wage ratio of two, the operator used the machine less than the optimal model. Every time the operator assigned the costly machine, so did the model, which is why the second performance measure is 1.0 at that point.

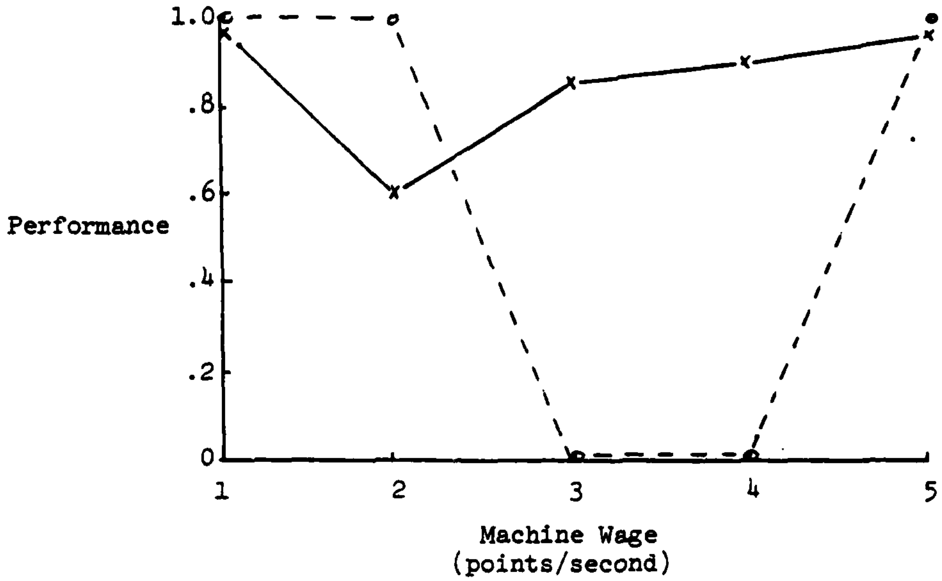

Figure 11.3: Operator performance as measured by the fraction of times the optimal model agrees with the subject's decisions for
—x— all decisions
--o-- only those decisions assigning the expensive machine

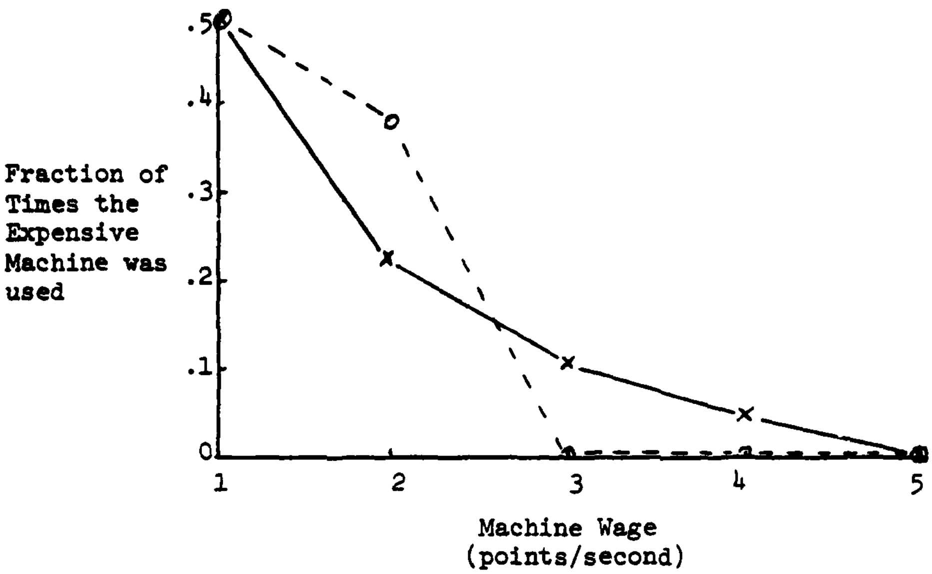

Figure 11.4: Fraction of all decisions to deal with a task where the expensive machine was used -- for decisions made by
—x— experimental subjects
- - o - - the optimal model

The impact of high wages on use of the expensive machine is plotted in Figure 11.4. Though the model actually reached zero uses of this machine faster than the operator, the operator was not far behind. The case where the model actually used the expensive machine more than the operator is shown at the wage ratio of two.

It is unclear why machine wage should have had a greater impact on human behavior than does productivity. Perhaps it was because during the Task Familiarization Phase the operator could see just how fast his score started to drop when he used an expensive machine. In addition, costs are more tangible to humans than are productivities and operators may have been more aware of the fact that expensive machines are "bad" than they were aware that low productivity machines were.

A system designer may be aware that high wages have a big impact on human behavior. Unfortunately, in most real life situations the operator is not really aware of machine costs. Maintenance, depreciation and capital interest charges are all counted towards machine costs but they are all intangible and do not directly impact on the operator. It is paradoxical that the parameters that effected the operator most are the ones that the operator don't see.

11.4 Conclusions

For the most part, subjects developed heuristic strategies that worked reasonably well in most situations. Mental effort was allocated primarily to those parameters that seemed to have a big effect on performance.

A general comment is that subjects liked to feel they were accomplishing something. Having many machines running simultaneously seemed to many subjects to by a good measure of performance. This goal seems to have increased the use of unproductive machines as was noted above. It was also noted that when a machine was very productive, but only on one work area, subjects would first check that work area in hopes of assigning a machine there before going on to other work areas. In that way the machine would be working if they found a task to do manually.

Subjects exhibited a bias towards the use of available machine aids. This bias did not affect the use of very productive machines because they should have been used when possible. However, this bias manifested itself in the sub-optimal use of unproductive machines.

CHAPTER 12

CONCLUSIONS AND SUGGESTED FURTHER RESEARCH

12.1 Conclusions From This Study

In this study, experimental subjects were placed in an environment where tasks appeared requiring service. Machine aids were provided to increase operator productivity. The operator in turn assumed a supervisory role searching out tasks and assigning them to machines. The operator could also get his hands dirty and complete a task himself, if he so chose.

Subjects performance in these tasks was compared to an optimal model. In the area of searching for tasks from work area to work area, subjects would resort to a fixed search pattern or apply a simple heuristic. Subjects could become very careless in deciding where to look next for tasks. However this carelessness was usually found only in those cases where added care would not have lead to improved performance. Where search strategy was important, subjects did put more effort into their search algorithm though they never reached the level of optimal behavior.

Subjects were also found to have an appreciable bias towards using machine aids when available. Not only would operators use machine aids that were very productive, they would use them even when the operator could do tasks much more efficiently himself.

A system designer worried about how efficiently operators are going to use any aids they are given should not be concerned with humans usurping jobs that should go to machines. Rather he should be concerned with the opposite. If the operator is supplied with a slow aid for use only when he is personally overloaded, he may instead the aid even when his work load is normal or below average.

Human operators can be inhibited from using machine aids if they are made aware of the true costs involved. Experiments showed that raising the wage paid to a machine dramatically reduced the fraction of times it was employed.

It should be noted here that though the experiments undertaken dealt with machine aids, the results are applicable to any sort of operator assistance. The co-pilot in an aircraft is nothing but a very complicated and autonomous aid to the pilot. In the broadest sense this report has looked at the general problem of assigning tools with varying productivities to tasks with varying requirements.

12.2 Possible Extensions on this Study

The most interesting results of this study dealt with the use of machine aids that either because of high wages or low productivity were poor choices to use in completing tasks. There are other aspects of machines that decrease their usefulness that were not considered here.

Another feature of machine productivity is the probability that the machine will successfully complete tasks it is assigned to. In this study a task assigned to a machine was as good as done. In many real life situations, however, machines break down or get stuck. The operator must assume the supervisory role of checking up on machines to make sure they are functional. Knowing that he will have to monitor machines will increase an operator's reluctance to undertake tasks manually.

Assigning machines in the experimental paradigm involved simply pressing a button. In reality machines may require a finite set up time above the actual task requirements. During this assignment process neither the machine nor the human are productively employed. Set-up time will have an effect on machine assignments.

A final modification to the paradigm used in this study would be not to have finite task queues. An implicit cost of leaving a task unattended in the present experimental situation is that the rewards of possible new tasks that might appear on the occupied work area are foregone. With infinite queues the costs of assigning a slow machine are reduced. Because subjects exhibit a bias to the overuse of machines their performance might improve even if behavior stays fixed.

CHAPTER 13

REFERENCES

[1] Albright, S. Christian. "Structure of Optimal Policies in Complex Queuing Systems" Operations Research Vol. 25, No. 2, November-December 1977.

[2] Bell, Colin E. "Turning Off a Server with Customers Present: Is This Any Way to Run an M/M/c Queue with Removable Servers ?" Operations Research Vol. 23. 1975.

[3] Carbonell, Jaime R. "A Queuing Model of Many-Instrument Visual Sampling" IEEE Transactions on Human Factors in Electronics Vol. HFE-7, No. 4, December 1966.

[4] Coombs, C. H.; Davies, R. M.; and Tversky, A. Mathematical Psychology. New Jersey: Prentice-Hall, Inc. 1970.

[5] Cox, D. and Smith, W. Queues . London: Meuthen. 1971.

[6] Freedy, A.; Davis, K. B.; Steeb, R.; Samet, M. G.; and Gardiner, P. C. Adaptive Computer Aiding in Dynamic Decision Processes: Methodology, Evaluation, and Applications. Contract N00014-73-C-0286, Engineering Psychology Programs, Office of Naval Research, Arlington, Virginia. August 1976.

[7] Fischoff, Baruch. "Debiasing" Judgement Under Uncertainty : Heuristics and Biases D. Kahneman, P. Slovic, and A. Tversky (Eds.). New York: Cambridge University Press. 1981.

[8] Govindaraj, T., and Rouse, William B. "Modeling the Human as a Controller in a Multitask Environment" Fourteenth Annual Conference on Manual Control, NASA Publication 2060. November 1978.

[9] Govindaraj, T. "Human Attention Allocation Strategies in Situations with Competing Criteria - Development of an Experimental Paradigm and Outlines for a Model" International Federation of Automatic Control Eighth World Conference Kyoto, Japan. August 1981.

[10] Harrison, J. Michael. "Dynamic Scheduling of a Multi-class Queue: Discount Optimality" Operations Research Vol. 23, No. 2, March-April 1975.

[11] Harrison, J. Michael. "A Priority Queue with Discounted Linear Costs" Operations Research Vol.23, No. 2, March-April 1975.

[12] Howard, Ronald A. Dynamic Programming and Markov Processes. New York: M.I.T. Press and John Wiley and Sons, Inc.. 1960.

[13] Kernighan, Brian W. and Plauger, P. J. Software Tools Ontario: Addison-Wesley Publishing Company. 1976.

[14] Johannsen, Gunnar. "Human-Computer Interaction in Decentralized Control and Fault Management of Dynamic Systems" International Federation of Automatic Control Eighth World Conference Kyoto, Japan. August 1981.

[15] Moray, Neville. Human Information Processing and Supervisory Control. Contract N00014-77-C-0256, Engineering Psychology Programs, Office of Naval Research, Arlington, Virginia. May 1980.

[16] Parsons, Henery McIlvaine. Man-Machien System Experiments. Baltimore: The John Hopkins Press. 1972.

[17] Rouse, William B. "Human-Computer Interaction in Multi-Task Situations" IEEE Transactions on Systems, Man and Cybernetics Vol. SMC-7, No. 5, May 1977.

[18] Schumacher, W., and Geiser, G. "Human Control Strategies in Concurrent Binary Tasks Under Overload Conditions" International Federation of Automatic Control Eighth World Conference Kyoto, Japan. August 1981.

[19] Sheridan, Thomas B. Towards a General Model of Supervisory Control. Cambridge, Massachusetts: M.I.T., working paper.

[20] Sheridan, Thomas B. and Ferrell, W. R., Man-Machine Systems. Cambridge, Massachusetts: M.I.T. Press. 1974.

[21] Sheridan, Thomas B. "On How Often the Supervisor Should Sample" IEEE Transactions on Systems, Science and Cybernetics. Vol. Scc-6, No. 2. April 1970.

[22] Smith, Wayne E. "Various Optimizers for Single-Stage Production" Naval Research Logistics Quarterly Vol. 3, No. 1, March 1956.

[23] Tcha, Dong-Wan and Pliska, Stanley R. "Optimal Control of Single-Server Queuing Networks and Multi-class M/G/1 Queues with Feedback" Operations Reseach Vol. 25, No. 2, March-April 1977.

[24] Tulga, Mehmet Kamil, and Sheridan, Thomas B. "Modeling Human Decision Making Behavior in Supervisory Control" Proceedings of the Seventh International Conference on Manual Control. Washington, D.C., September 1977.

[25] Tulga, Mehmet Kamil. Dynamic Decision Making in Multitask Supervisory Control: Comparison of an Optimal Algorithm to Human Behavior. Sc.D. Thesis, Massachusetts Institute of Technology. September 1978.

[26] Tversky, Amos and Kahneman, Daniel. "Judgement under Uncertainty: Heuristics and Biases" Science. Vol. 185, 27 September 1974.

[27] Tversky, Amos and Kahneman, Daniel. "The Framing of Decisions and the Psychology of Choice" Science. Vol. 211, 30 January 1981.

APPENDIX A

MATHEMATICS OF POISSON PROCESSES

A.1 Poisson Processes

Many arrival processes are assumed to have Poisson distributions. The key feature of a Poisson distribution is that the likelihood of an event occuring in some small time interval is exactly equal to the likelihood of an occurance in any other equally small time interval. There is no memory in the system. The instantaneous probability that an event will occur is independent of system history.

If a commuter arrives at a bus stop with no information about the bus schedule, he has no reason to expect the bus to be more likely to arrive in exactly ten minutes than to expect it to arrive immediately. Similarly, a machine is likely to experience breakdowns that appear to have no deducable cause. Both bus arrivals and machine breakdowns can be modeled as having a Poisson distribution.

Poisson distributions are characterized by the parameter L, the mean rate of event occurances. The probability that k events will occur within a time interval t is given by:

$$\Pr[k \text{ events in } t] = p_k(t) = \frac{(L*t)^k}{k!} \exp(-L*t) \qquad (A.1)$$

A.2 The Exponential Distribution

Another concern in the study of Poisson processes is the distribution of interarrival times between events. The interarrival time is the period after one event and before the next during which nothing happens. The interarrival time for Poisson processes has an Exponential distribution

with the same defining parameter L.

$$EX[t]=L*\exp(-L*t) \tag{A.2}$$

The probability that an event will occur at some time within an interval T is simply the integral of EX[t] over all values of t between 0 and T. This probability of occurance is signified by PO(T,1), or by PO(T) in other sections of this paper.

$$PO(T,1) = PO(T) = \int_0^T EX[t]\, dt$$

$$PO(T) = 1 - \exp(-L*T) \tag{A.3}$$

A.3 Formulae with Holding Cost

In certain sections of this paper tasks with Exponential interarrival times exact a holding cost h for each unit of time they are unattended. Consider the following related scenario. At time 0, no events have occured and an operator turns his attention elsewhere. At time T he notices that an event has indeed occured as some past time. What is the expected net cost of this task up to time T ?

The expected cost of an event over time T, given that the event has indeed occured, is simply the integral over all values of t between 0 and T of the cost of the event occuring at t [i.e. h*(T-t)] times the probability that the event actually occured at t. This quantity is normalized by the probability that the event did occur in T.

Expected cost given event occurance, $E_{c|o}(T,L)$:

$$E_{c|o}(T,L) = \frac{\int_0^T EX(t)*h*(T-t)\, dt}{PO(T,L)}$$

$$E_{c|o}(T,L) = \frac{(h/L)*[L*T+\exp(-L*T)-1]}{1 - \exp(-L*T)} \tag{A.4}$$

APPENDIX B

LISTINGS OF COMPUTER PROGRAMS

The programming and experiments for this study were conducted on a PDP 11/34 in the Man-Machine Systems Laboratory at the Massachusetts Institute of Technology. The visual interface with subjects was provided by a Megatek 7000 Vector Graphic Display.

The computer language RATFOR was used for most of the programming. RATFOR is a language based on FORTRAN, but which adds modern control structures including IF-ELSE, WHILE, and REPEAT-UNTIL. RATFOR is considered by its creators to be a RATional FORtran; hence its name. Explanations of both the design and structure of RATFOR programs can be found in <u>Software Tools</u> by Kerighan and Plaugher (Addison-Wesley, Don Mills, Ontario, 1976).

The following is an alphabetized list of major programs used in this study plus brief descriptions of their functions. Actual listings of these programs are available from the Man-Machine Systems Laboratory at M.I.T..

BINARY: Converts a decimal number into its binary equivalent (used in TCST)

BOXCLR: This program places actual experimental subjects in the environment of certainty described in Ch. 6

BSETUP: Sets up the series of 96 experiments with BOXCLR described in Ch. 6

BXPERF: This program takes operator strategies from BOXCLR as specified by BSETUP and compares them to

all possible strategies in order to generate performance mearsures

CHKPOL: Compares actual operator decisions from SUPER to the N-step model presented in Ch. 7

CMMND: Takes operator commands from the Contol Box shown in Ch. 5 (used in SUPER and BOXCLR)

DISPLY: Displays Score and Time of and experiment on the Megatek display (used in SUPER and BOXCLR)

DRAWI: A system subroutine that draws a line on the Megatek dislay to to point (x,y)

DRWFLD: Draws the SUPER playing field on the Megatek display (used in SUPER)

DRWMAC: Draws machine aids on the Megatek display (used in SUPER)

DRWSCR: Draws the outlines for the SUPER Megatek display (used in SUPER)

DRWTXT: Draws most of the text for SUPER experiments on the Megatek display (used in SUPER)

ECO: Computes the expected holding cost incurred by not looking at a work area for T seconds, assuming a task does appear there (used in GEN)

GEN: Generates the transition probabilities, reward matrix, and state transition times for the Markov model in Ch. 5 (used in RPTH2W)

HEADER: Places a header showing time, date and a title on the output files from experiments (used in SUPER)

LDTRN3: A system subroutine which translates and scales all subsequent commands in the Megatek display buffer

HSTGRM: Computes the histogram of the costs of all possible strategies for BOXCLR (used in BXPERF)

MGSEND: A system subroutine which sends a set of Megatek commands to the Megatek display buffer

MOVEI: A system subroutine that moves the current position of the Megatek display sequence to the point (x,y)

MPRD: Takes the product of two 4x4 matrices (used in RPTH2W)

ORD: Orders the elements of an array in a monatonically increasing sequence (used in BSETUP and BXPERF)

PO: Computes the probability a task will appear within T second (used in GEN)

POLCY: Determines the best decision for the N-step model presented in Ch. 7. A seperate version of POLCY is required for each level of the N-step search. A 3-level example is included in this appendix (used in STRATP and CHKPOL)

RDRAWI: A system subroutine that draws a line on the Megatek display to a point relative to the current point in the display list

RMOVEI: A system subroutine that performs a relative move to the current point in te Megatek display buffer

RPTH2W: Computes the reward per unit time for the two-work-area / no-machine Markov model as a function of the critical transfer probability, PC (used in XH2W)

RPTMNW: Computes the reward per unit time for the N-work-area / M-machine model through simulation (used in XMNW)

SETINT: A system subroutine that sets the intensity of subsequent lines drawn on the Megatek display

STRATP: Simulates an operator using N-step model described in Ch. 7 (used in RPTMNW)

SUPER: This program places an experimental subject in a supervisory role over machines. Tasks appear in different work areas and must be dealt with. SUPER is described more fully in Ch. 5

TCST: Computes the total cost of a strategy for BOXCLR as described in Ch. 6 (used in HSTGRM and BXPERF)

XH2W: Finds the optimal value of PC in the Markov model presented in Ch. 5

XMNW: Explores the case of M machines and N work areas through simulation, as described in Ch. 7

DISTRIBUTION LIST

CDR Paul R. Chatelier
Office of the Deputy Under
Secretary of Defense
OUSDRE (E&LS)
Pentagon, Room 3D129
Washington, D.C. 20301

Engineering Psychology Programs
Code 442
Office of Naval Research
800 North Quincy Street
Arlington, VA 22217 (5 cys)

Undersea Technology Programs
Code 220
Office of Naval Research
800 North Quincy Street
Arlington, VA 22217

Dr. Eugene Silva
Ocean Sciences, Code 421
Office of Naval Research
800 North Quincy Street
Arlington, VA 22217

Communication & Computer Technology
Programs
Code 240
Office of Naval Research
800 North Quincy Street
Arlington, VA 22217

Manpower, Personnel, and Training
Programs
Code 270
Office of Naval Research
800 North Quincy Street
Arlington, VA 22217

Information Sciences Division
Code 433
Office of Naval Research
800 North Quincy Street
Arlington, VA 22217

Psychology & Neuro Biology Programs
Code 441B
Office of Naval Research
800 North Quincy Street
Arlington, VA 22217

Special Assistant for Marine
Corps Matters
Code 100M
Office of Naval Research
800 North Quincy Street
Arlington, VA 22217

Commanding Officer
ONR Eastern/Central Regional Office
ATTN: Dr. J. Lester
Building 114, Section D
666 Summer Street
Boston, MA 02210

Commanding Officer
ONR Western Regional Office
ATTN: Mr. R. Lawson
1030 East Green Street
Pasadena, CA 91106

Dr. Perry Alers
Marine Systems Branch, Code 5821
Naval Research Laboratory
Washington, D.C. 20375

Director
Naval Research Laboratory
Technical Information Division
Code 2627
Washington, D.C. 20375

Dr. Michael Melich
Communications Sciences Division
Code 7500
Naval Research Laboratory
Washington, D.C. 20375

Office of the Chief of Naval Operations
Deep Submergence Systems Division
OP-231
Washington, D.C. 20350

Dr. Robert G. Smith
Office of the Chief of Naval
Operations, OP 987H
Personnel Logistics Plans
WASHINGTON, D.C. 20350

CDR G. Worthington
Office of the Chief of Naval
Operations, OP 372G
Washington, D.C. 20350

Dr. W. Mehuron
Office of the Chief of Naval Operations,
OP 987
Washington, D.C. 20350

Dr. Andreas B. Rechnitzer
Office of the Chief of Naval
Operations, OP 952F
Naval Oceanography Division
Washington, D.C. 20350

Dr. Jerry C. Lamb
Combat Control Systems
Naval Underwater Systems Center
Newport, RI 02840

Human Factors Department
Code N215
Naval Training Equipment Center
Orlando, FL 32813

Mr. John Freund
NAVSEA 05R2
Naval Sea Systems Command
Washington, D.C. 20362

Dr. Gary Poock
Operations Research Department
Naval Postgraduate School
Monterey, CA 93940

Mr. H. Talkington
Ocean Engineering Department
Naval Ocean Systems Center
San Diego, CA 92152

Mr. Paul Heckman
Naval Ocean Systems Center
San Diego, CA 92152

Mr. Warren Lewis
Human Engineering Branch
Code 8231
Naval Ocean Systems Center
San Diego, CA 92152

Dr. Ross L. Pepper
Naval Ocean Systems Center
Hawaii Laboratory
P. O. Box 997
Kailua, HI 96734

Dr. A. L. Slafkosky
Scientific Advisor
Commandant of the Marine Corps
Code RD-1
Washington, D.C. 20380

Naval Material Command
NAVMAT 0722 - Room 508
800 North Quincy Street
Arlington, VA 22217

Mr. Phillip Andrews
Naval Sea Systems Command
NAVSEA 0341
Washington, D.C. 20362

Mr. Milon Essoglou
Naval Facilities Engineering Command
R&D Plans and Programs
Code 03T
Hoffman Building II
Alexandria, VA 22332

CDR Robert Biersner
Naval Medical R&D Command
Code 44
Naval Medical Center
Bethesda, MD 20014

Dr. Arthur Bachrach
Behavioral Sciences Department
Naval Medical Research Institute
Bethesda, MD 20014

CDR Thomas Berghage
Naval Health Research Center
San Diego, CA 92152

Dr. George Moeller
Human Factors Engineering Branch
Submarine Medical Research Lab
Naval Submarine Base
Groton, CT 06340

Dr. Robert Blanchard
Navy Personnel Research and
Development Center
Command and Support Systems
San Diego, CA 92152

Mr. J. Williams
Department of Environmental Sciences
U.S. Naval Academy
Annapolis, MD 21402

Human Factor Engineering Branch
Naval Ship Research and Development
Center, Annapolis Division
Annapolis, MD 21402

Dr. James McGrath
CINCLANT FLT HQS
Code 04E1
Norfolk, VA 23511

Mr. John Quirk
Naval Coastal Systems Laboratory
Code 712
Panama City, FL 32401

CDR W. Moroney
Code 55MP
Naval Postgraduate School
Monterey, CA 93940

Dr. Joseph Zeidner
Technical Director
U.S. Army Resaerch Institute
5001 Eisenhower Avenue
Alexandria, VA 22333

Director, Organizations and Systems
Research Laboratory
U.S. Army Research Institute
5001 Eisenhower Avenue
Alexandria, VA 22333

Technical Director
U.S. Army Human Engineering Labs
Aberdeen Proving Ground, MD 21005

U.S. Air Force Office of Scientific
Research
Life Sciences Directorate, NL
Bolling Air Force Base
Washington, D.C. 20332

Chief, Systems Engineering Branch
Human Engineering Division
USAF AMRL/HES
Wright-Patterson AFB, OH 45433

Director, Human Factors Wing
Defence & Civil Institute of
Environmental Medicine
Post Office Box 2000
Downsview, Ontario M3M 3B9
CANADA

Defense Technical Information Center
Cameron Station, Bldg. 5
Alexandria, VA 22314 (12 cys)

Dr. Craig Fields
Director, System Sciences Office
Defense Advanced Research Projects
Agency
1400 Wilson Boulevard
Arlington, VA 22209

Dr. M. Montemerlo
Human Factors & Simulation
Technology, RTE-6
NASA HQS
Washington, D.C. 20546

Dr. J. Miller
Florida Institute of Oceanography
University of South Florida
St. Petersburg, FL 33701

Dr. Robert R. Mackie
Human Factors Research, Inc.
5775 Dawson Avenue
Goleta, CA 93017

Dr. H. McI. Parsons
Human Resources Research Office
300 N. Washington Street
Alexandria, VA 22314

Dr. Jesse Orlansky
Institute for Defense Analyses
400 Army-Navy Drive
Arlington, VA 22202

Dr. T. B. Sheridan
Department of Mechanical Engineering
Massachusetts Institute of Technology
Cambridge, MA 02139

Dr. Harry Snyder
Department of Industrial Engineering
Virginia Polytechnic Institute and
State University
Blacksburg, VA 24061

Dr. Robert T. Hennessy
NAS - National Research Council
2101 Constitution Ave., N.W.
Washington, DC 20418

Dr. Amos Freedy
Perceptronics
6271 Variel Avenue
Woodland Hills, CA 91364

Dr. Robert Williges
Human Factors Laboratory
Virginia Polytechnic Institute
and State University
130 Whittemore Hall
Blacksburg, VA 24061

Dr. Alphonse Chapanis
Department of Psychology
The Johns Hopkins University
Charles and 34th Streets
Baltimore, MD 21218

Dr. James H. Howard, Jr.
Department of Psychology
Catholic University
Washington, D.C. 20064

Dr. Christopher Wickens
University of Illinois
Department of Psychology
Urbana, IL 61801

Dr. Edward R. Jones
Chief, Human Factors Engineering
McDonnell-Douglas Astronautics
Company
St. Louis Division
Box 516
St. Louis, MO 63166

Dr. Babur M. Pulat
Department of Industrial Engineering
North Carolina A&T State University
Greensboro, NC 27411

Dr. Richard W. Pew
Information Sciences Division
Bolt Beranek & Newman, Inc.
50 Moulton Street
Cambridge, MA 02138

Dr. Douglas Towne
University of Southern California
Behavioral Technology Laboratory
3716 S. Hope Street
Los Angeles, CA 90007

Dr. A. K. Bejczy
Jet Propulsion Laboratory
California Institute of Technology
Pasadena, CA 91125

Dr. William R. Uttal
Institute for Social Research
University of Michigan
Ann Arbor, MI 48109

LMED

8